Universidad
Carlos III de Madrid
www.uc3m.es

TESIS DOCTORAL

An Ontology for Human-Like Interaction Systems

Autora:

Esperanza Albacete García

Director/es:

Francisco Javier Calle Gómez

Elena Castro Galán

DEPARTAMENTO DE INFORMÁTICA

Leganés, Enero de 2016

TESIS DOCTORAL

An Ontology for Human-Like Interaction Systems

Autor: *Esperanza Albacete García*

Director/es: Francisco Javier Calle Gómez

Elena Castro Galán

Firma del Tribunal Calificador:

		Firma
Presidente:	Inés Mª Galván León	
Vocal:	Yolanda García Ruíz	
Secretario:	José Mª Cavero Barca	

Calificación:

Leganés/Getafe, de de

Contents

1 Introduction

This report proposes and describes the development of a Ph.D. Thesis aimed at building an ontological knowledge model supporting Human-Like Interaction systems. The main function of such knowledge model in a human-like interaction system is to unify the representation of each concept, relating it to the appropriate terms, as well as to other concepts with which it shares semantic relations.

When developing human-like interactive systems, the inclusion of an ontological module can be valuable for both supporting interaction between participants and enabling accurate cooperation of the diverse components of such an interaction system. On one hand, during human communication, the relation between cognition and messages relies in formalization of concepts, linked to terms (or words) in a language that will enable its utterance (at the expressive layer). Moreover, each participant has a unique conceptualization (ontology), different from other individual's. Through interaction, is the intersection of both part's conceptualization what enables communication. Therefore, for human-like interaction is crucial to have a strong conceptualization, backed by a vast net of terms linked to its concepts, and the ability of mapping it with any interlocutor's ontology to support denotation.

On the other hand, the diverse knowledge models comprising a human-like interaction system (situation model, user model, dialogue model, etc.) and its interface components (natural language processor, voice recognizer, gesture processor, etc.) will be continuously exchanging information during their operation. It is also required for them to share a solid base of references to concepts, providing consistency, completeness and quality to their processing.

Besides, humans usually handle a certain range of similar concepts they can use when building messages. The subject of similarity has been and continues to be widely studied in the fields and literature of computer science, psychology and sociolinguistics. Good similarity measures are necessary for several techniques from these fields such as information retrieval, clustering, data-mining, sense disambiguation, ontology translation and automatic schema matching. Furthermore, the ontological component should also be able to perform certain inferential processes, such as the calculation of semantic similarity between concepts. The principal benefit gained from this procedure is the ability to substitute one concept for another based on a calculation of the similarity of the two, given specific circumstances. From the human's perspective, the procedure enables referring to a given concept in cases where the interlocutor either does not know the term(s) initially applied to refer that concept, or does not know the concept itself. In the first case, the use of synonyms can do, while in the second one it will be necessary to refer the concept from some other similar (semantically-related) concepts.

Therefore, processing similarities is not only a matter of *synonymy*, but also take part in such process other semantic relationships like *hyperonymy*/*hyponymy*, *holonymy*/*meronymy*, antonymy, foreignness, etc. The more semantic links the ontology observes (the stronger it is) the more accurate will be the similarity calculations it supports. A really strong ontology should even help to the processing of complex figures, such as polysemy/homonymy, paronymy, epithet, metaphor, metonymy, etc. Similarity helps in the resolution of complex figures as metaphor due to direct semantic relationships.

During human interaction, there occur often quick travels from denoted concepts to understood concepts, performed by means of similarity links. This concept mutation empowers understanding during human interaction, which utterly involves more communication abilities with less knowledge. Eventually, the receiver will have slightly lower confidence on interpretations based on mutated concepts but, while reaching a threshold, it will be usually enough to trigger the proper consequences (*perlocutions*). For example, if any tree is assumed to be taken to the sawmill and the interlocutor provides an oak, it will be taken there despite of the lack of specific rules for that kind of tree. In case the degree of confidence in the interpretation is not enough, the interlocutor will seek further reinforcement, but counting on some interpretation enables less costly reinforcement techniques (implicit or explicit reaffirmation, …) that taking no interpretation at all (interruption).

Let's pose another example. When characterizing users, an interlocutor who stated being forty years old may be matched with a stereotype of 'mature age' to some extent (even also matched with another group of 'young age' with other certainty value). Mismatches with clearly understood concepts can help to unveil rhetoric figures and tropes. Then, mutated concepts do not only help to fix the accurate reference, but can also help to choose the proper pattern for the whole sentence during the natural language interpretation, supporting better understanding of each intervention and consequently of the entire dialogue.

Besides, semantic similarity also enables the system building explanations about message meaning. Providing them to the user should help to clarify a given misunderstood concept based on similar concepts, thereby enhancing communicative effectiveness. First strategy can be based on the use of reformulated utterances and synonyms (if any). But many cases would reveal the lack of the focused concept (or synonyms are not appropriate). Then, the system can build explanations, which may include (but is not restricted to) hyperonyms, hyponyms, cohyponyms, holonyms, meronyms, antonyms, foreign terms, etc. But what is really challenging is focusing this feature from the opposite point of view: the system should be able of understanding such explanations regarding previously-unknown semantic relations between known concepts, formalize such knowledge and hoard it for further use. Moreover, the system should be even able of acquiring new (unknown) concepts, as long as the user is able to relate them to similar concepts that are previously known by the system. This mechanism can benefit to different knowledge models. For example, the user model of a human-like interaction system will be able to take advantage of the similarity between an acquired characteristic of a user and one characterizing a user group (e.g., a 31-year-old user is very similar to a 30-year-old user even if the two are related to different concepts, 31-year-old and 30-year-old, for the same feature). It also improves the effectiveness of the context model which, by describing the circumstances of the interaction, allows other models to access a subset of knowledge that is relevant in those circumstances. For example, if the dialog model does not find strategies applicable to the circumstance 'conference', it can apply the strategies available to another circumstance 'teach' given the similarities between these two concepts in the ontology. Thus, the challenge is to develop a dialogue system able of learning new concepts and semantic relations by itself, enriching its ontology for improving future interactions.

What usually hinders the development of such ontologies is their maintenance, because the related mechanisms are usually too costly. Specifically, feeding these ontologies, a task that is usually performed manually by experts, involves unbearable costs. The fore proposed challenge, to develop advanced mechanisms for automatic knowledge acquisition, appears to be a proper solution: knowledge is obtained through thousands (or more) of human-like interactions with human subjects. Actually, this procedure is the way humans incorporate new concepts and terms to their knowledge (through both oral and written communication). Consequently, the goal is to imitate human behavior to attain a sustainable solution. The referred human behavior involves some supporting concepts and process that have to be taken into account, such as reputation and reliability. The initial reliability on the knowledge will depend upon the trust on the source (which may vary during time) and its reputation on a given subject (facts from the same source could be taking different initial reliability if are referred to different domains). Anyhow, of course, any knowledge's reliability is subject to review due to feedbacks produced through its use. Summarizing, through the system lifetime, the knowledge bases would be enriched by interacting with the users, thus learning new concepts, terms and relationships, and by continuous refinement of their knowledge. Despite this ontology proposal requires vast amount of knowledge which acquisition would entail unbearable costs, the knowledge crowdsourcing focus may overcome the obstacle, turning the approach into a realistic solution.

This document is structured as follows: in the first place, Section 2 reviews the literature on theoretical foundations of ontologies which involves the main languages and representation tools, similarity measures and ontology learning techniques. Section 3 presents the main objectives and the proposal of the Ph.D. thesis. Sections 4 and 5 contain the evaluation techniques and the discussion, respectively. Finally, last section includes some conclusions and challenges.

2 Theoretical Foundations of Ontologies

Ontologies are widely used in areas such as Knowledge Engineering, Artificial Intelligence and Computer Science, in applications related to knowledge management, natural language processing, e-commerce, information retrieval, database design and integration, bio-informatics and in other emerging fields. However, the term 'Ontology' applied to computer science often refers to different sort of technology, and this sort of polysemy may lead to confusion. From an ontological point of view, the term has a long development in the field of philosophy, in which it refers to the subject of existence. Philosophers of language refer to it as answering "What is", regarding the usage of terms as vehicles to communicate concepts in messaging. However, the term 'Ontology' is often with that other of epistemology, which is about knowledge and knowing, and therefore referring the knowledge enclosed under terms within a specific knowledge domain.

2.1 The origin of the term 'Ontology'

The term 'Ontology' applied to human-like interaction systems is a concept resulted from an evolution of meaning that is shown in Figure 1. Several disciplines framed in different areas such as philosophy, linguistic and engineering have contributed to the meaning of this concept.

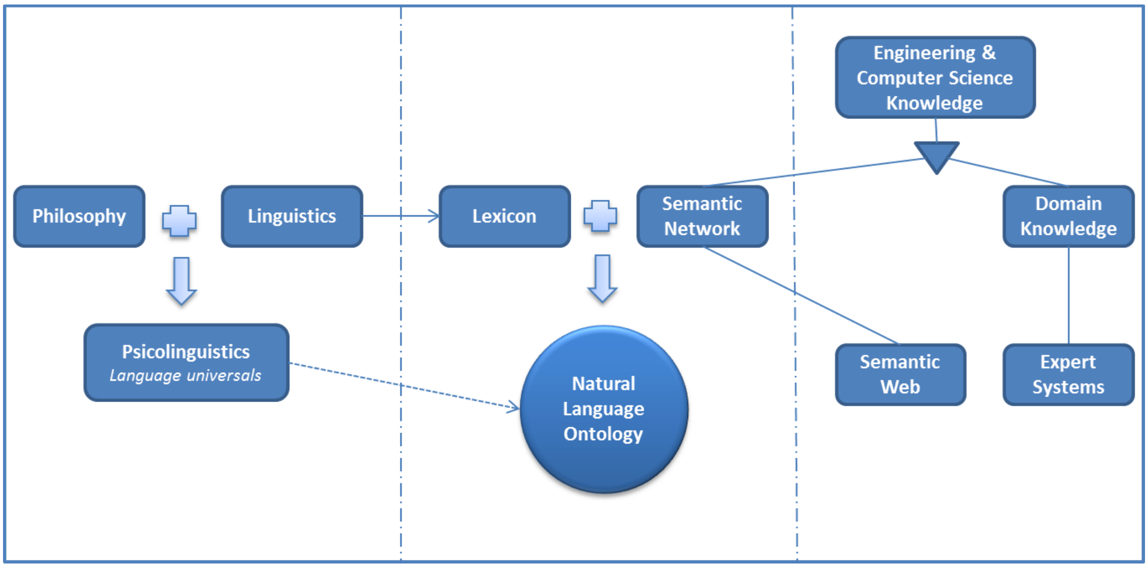

Figure 1 – Evolution of the concept Natural Language Ontology

As previously stated, in the philosophical field, the word 'ontology' stands for the study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations. It is intended to answer questions like "What are the things?", "What is the essence within things?" (even when they change their colour, size, etc.) or "How can entities of the world be classified?" (taxonomy). In the last two decades, this word has become relevant for the Knowledge Engineering community. In this field, the term refers to a knowledge model restricted to a specific domain (often supporting an Expert System). Therefore, the questions it is intended to be answering are more or less similar, but adding the tagline "within my domain". Sometimes its scope is restricted to a specific purpose within its domain, in which case it would be more appropriate to be referring as "specific purpose knowledge model". Until now, a large number of ontologies have been developed by different experts, under different approaches, and with different methods and techniques. 'Ontological Engineering' refers to the set of activities concerning the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies.

Meanwhile, philosophers of language consider primarily linguistic data and introspection for drawing generalizations to be used as conceptualizations for building messages. They are found in two main trends: seeking universals of language, where the Ontology is the shared gathering of all conceptualizations, partially known by any speaker; or rejecting universals existence, by identifying personal ontologies for each culture (even each speaker) and the ability of connecting (or mapping) them through interaction. Traditional lexical semanticists mainly use lexical resources as a ground for that conceptualization. Cognitive scientists might broaden the range of information sources, by including other perceptual modes such as visual or tactile information. Anyhow, in these approaches, Ontology can be seen as the cognitive foundation on which to build human interaction.

Regarding Human Computer Interaction, several definitions of the word 'ontology' have been provided by different authors. All of them offer different and complementary points of view on the same reality. Some authors provide definitions that are independent of the processes followed to build the ontology and of its use in applications, while other definitions are influenced by its developments process. The definition most quoted in literature is the Gruber's one: "An ontology is an explicit specification of a conceptualization." where a conceptualization is basically the idea of the world that a person or a group of people can have. Explicit specification of conceptualization means that Ontology is a description (like a formal specification of a program) of the concepts and relationships between them that may exist for an agent or a community of agents.

2.2 Conceptualization vs. specification

The conceptualization is the relevant informal knowledge one can extract and generalize from experience, observation, and introspection. A conceptualization is relevant information itself and it is independent from specific situations or representation languages (see Figure 2). The specification is the encoding of the conceptualization in a representation language. To be useful, a conceptualization has to be shared among agents, such as humans. In other words, the conceptualization that natural language represents is a collective process, not an individual one and the information content is defined by the collectivity of speakers.

An ontology implementation does not have to express all the possible constraints. The level of details in conceptualization depends on the requirements of the intended application and expressing conceptualization in ontology depends on the used ontology language. On the one hand, intended models refers to those involving the description of the domain (what is possible within it); and in the other hand, ontology models are a restriction of the possible models that express conceptualization.

From the knowledge engineering, ontologies are important for the purpose of enabling knowledge sharing and reuse. An ontology is in this context a specification used for making ontological commitments. Practically, an ontological commitment is an agreement to use a vocabulary (i.e., ask queries and make assertions) in a way that is consistent (but not complete) with respect to the theory specified by an ontology. Agents then commit to ontologies and ontologies are designed so that the knowledge can be shared among these agents.

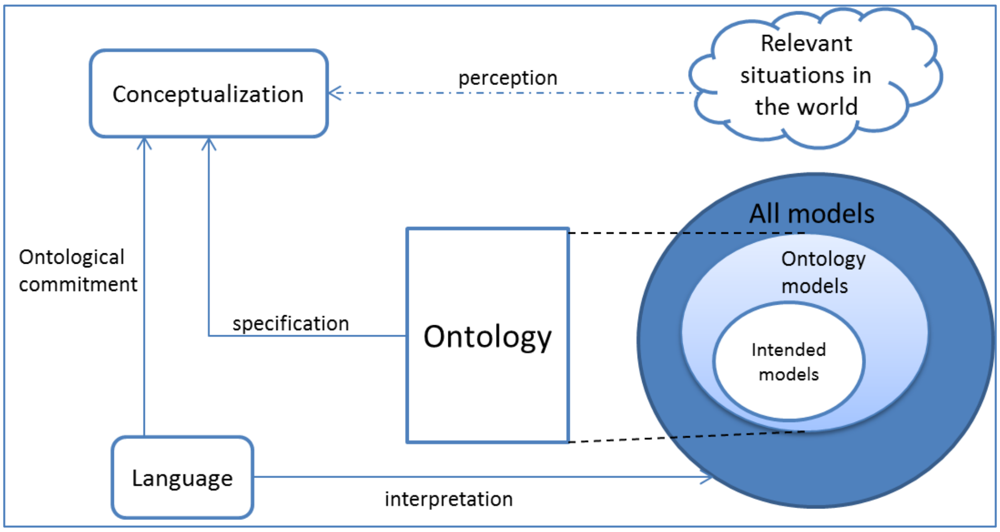

Figure 2 - Relationships between conceptualization, specification, language and ontology

All agents, whatever their commitment to an ontology is, find themselves in a communication situation illustrated using the semiotic triangle (see Figure 3). The communicator of a message may use a word or more generally, a sign like the phoneme of the string "bicycle" to stand for a concept the sender has in his own mind. He uses the sign in order to refer to abstract or concrete things in the world, which may, but need not be, physical objects. This agent also invokes a concept in the mind of an actor receiving this sign. The receiver uses the concept in order to point out the individual or the class of individuals the sign was intended to refer to. Thereby, the interpretation of the sign as a concept as well as its use in a given situation depends heavily on the receiver as well as the overall communication context.

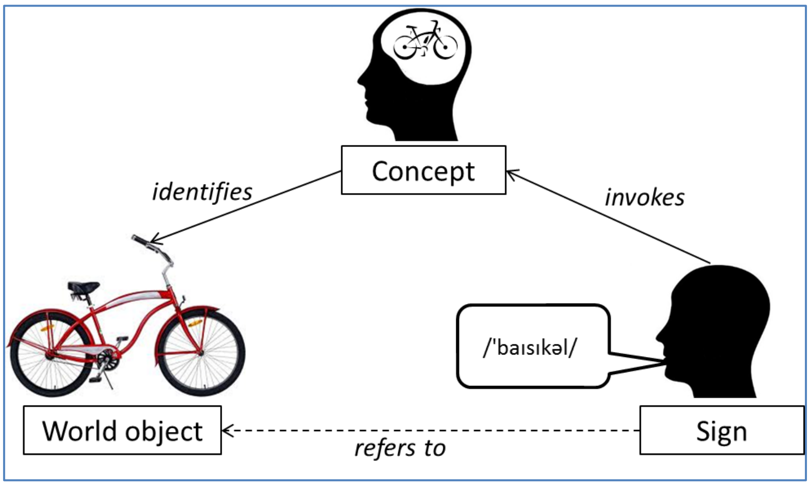

Figure 3 - Semiotic triangle

3 State of the Art

This section provides a general perspective of the state of the art of ontologies that includes in the first place the different types of ontologies according to different classifications (Section 3.1). Secondly, the state of the art will be focused in linguistic ontologies (Section 3.2) where their applications will be discussed and some examples will be described. Section 3.3 presents the main methodologies, representation languages and tools for building ontologies. The following subsection (3.4) includes the most relevant related work about semantic similarity measures and, in Section 3.5 the state of the art will be focused in the different techniques to automatically learn ontological knowledge. Finally, the conclusions of the state of the art that underlie the proposal of this thesis are presented in Section 3.6.

3.1 Types of Ontologies

This section analyses the different families and types of ontology models in order to compare them and detect their common characteristics and differences. This analysis will provide the context in which the lexical ontologies are placed and will show how each type is related to the rest.

There are different types of families of ontologies depending on the criteria used to classify them: their structure, their content or their degree of formalism.

3.1.1 Ontology Classification based on the Structure

Ontologies are usually classified according to the amount and type of structure of their conceptualization. Following this criteria, Sowa (2000) distinguishes two main families: the lexical or terminological ontologies and the formal or axiomatized ontologies.

- **Lexical or terminological ontologies:** These ontologies have concepts and relations that are not fully specified by axioms and definitions that determine the necessary and sufficient conditions of their use. Their concepts may be partially specified by relations such as *is-a*, or part-whole. These types of relations only determine the relative positions of the concepts with respect to another, but do not completely define them.
- **Formal or axiomatized ontologies:** They are lexical ontologies whose concepts and relations have associated axioms and definitions that are defined in formal logic or in some computer-oriented language that can be translated to logic. Formal ontologies tend to be smaller than lexical ones, but their axioms can support more complex inferences and computations.

Lexical ontologies can also be defined using formal logic, but this logic is usually simpler, less expressive and more easily computable than full first-order logic predicates used to define formal ontologies. Actually, the difference between the two families of ontologies, lexical and formal, is the complexity rather than kind. Lexical or terminological ontologies do not have enough expressivity to represent the relationship complexity required by many applications; however it is easier to create an integration with them than with formal ones. With this in mind, the selection of a lexical or a formal ontology depends exclusively on the required functionality.

Another classification based on the structure was defined by (Van Heijst, Schreiber, & Wielinga, 1997). Their proposal divides ontologies into the following three types:

- **Terminological ontologies:** this type specifies the terms that are used to represent knowledge in the domain of discourse. This terminological definition would be equivalent to Sowa's lexical/terminological one.
- **Knowledge modelling ontologies:** these ontologies are used to specify conceptualizations of the knowledge. This type of ontologies would fit into the Sowa's formal/axiomatized class.
- **Information ontologies:** specify the terms that are used to represent the record structure of databases. This type would lie somewhat in the middle between Sowa's lexical and formal ontological types because they have many of the features of the formal ones but lack some key elements of terminological ones (for example, *is-a* relations).

3.1.2 Ontology Classification based on the Content

Ontologies can also be classified based on their subject of conceptualization, i.e., their content. (Guarino, 1998) defined the following three classes:

- **Top-level ontologies:** This type of ontologies describes very general concepts which are independent of a particular problem or domain. Some of them are, for example, time, space, object, event or action).
- **Domain and Task ontologies:** These ontologies describe the vocabulary related to a generic domain (like medicine or chemistry) or a generic task or activity (for example, selling or diagnosing). This conceptualization is made by specializing the terms introduced in the top-level ontology.
- **Application ontologies:** This type of ontologies contains all the definitions that are needed to model the knowledge required for a particular application. Application ontologies are usually specializations of domain or task ontologies because they usually describe concepts depending on a particular domain or task.

Another content classification, quite similar to Guarino's one was made by (Mizoguchi, Van Welkenhuysen, & Ikeda, 1995). They divided the ontologies into three types: general/common ontologies (similar to Guarino's top-level ones), domain ontologies and task ontologies.

Van Heijst et al., whose structure-based classification was previously mentioned, also defined a classification based on the content which includes: application ontologies, domain ontologies, generic ontologies (or top-level) and representation ontologies. This last one type describes the conceptualizations about knowledge representation formalisms, which are intended to be neutral with respect to world entities.

3.1.3 Ontology Classification based on the Degree of Formalism

(Lassila & McGuinness, 2001) proposed a classification of ontologies types according to the degree of formalism and semantics provided in their specification. They range from simple controlled vocabulary to complex reasoning models. In the Figure 4 is shown the categorization of ontologies proposed by Lassila and MacGuinness and the correspondence with the Sowa's classification in formal and terminological ontologies.

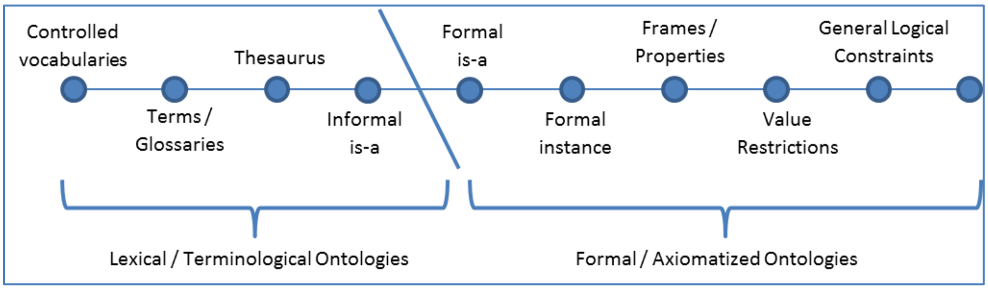

Figure 4 - Lassila and MacGuinness Ontologies Classification

Another ontology classification based on the degree of formalism was defined by (Sigel, 2006). This classification divides the ontologies focusing on the semantic interoperability point of view. As is shown in Figure 5, there are three ranges of interoperability:

- **Syntactic interoperability:** In this level are the taxonomies which are only able to express few semantics (*is-a* relationship).
- **Structural interoperability:** In this level are the thesauruses, which provide syntactic and structural interoperability. Terminological models defined previously correspond with these models with syntactic and structural interoperability.
- **Semantic interoperability:** this degree of interoperability is provided by logical theory models, and thanks to their strong semantics, provide the most complete for of semantic interoperability.

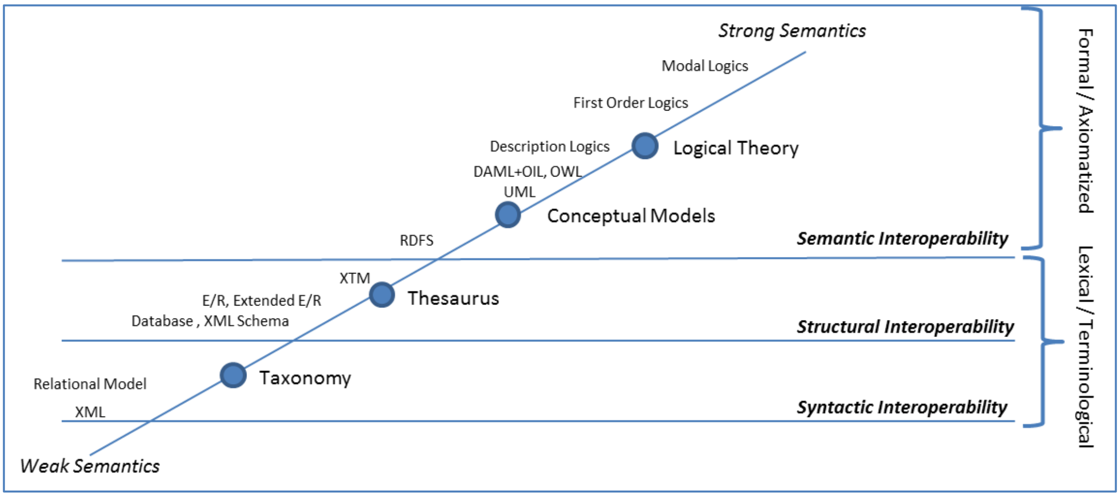

Figure 5 - Sigel Ontologies Classification

3.2 Linguistic Ontologies

In linguistic ontologies, conceptualization is based on linguistic criteria, more precisely information found in lexical resources such as dictionaries or thesauruses. In many cases they are slightly hybrid since though they feature mainly linguistic knowledge they also include some non-linguistic knowledge (encyclopaedic knowledge), such as names of places. Lexical ontologies are interesting because of the special status of the collection of linguistically conventionalized concepts in human cognition and they offer a heterogeneous amount of resources, used mostly in natural language processing. The main characteristic of these ontologies is that it's a linguistic function. Some linguistic ontologies suit grammatical units, others are based on lexical items, and others are cognitive.

Most linguistic ontologies use word as grammatical units, only a few like the Generalized Upper Model (GUM) (Bateman, Hois, Ross, & Tenbrink, 2010) or SENSUS (Knight & Luk, 1994) gather information on grammatical units that are bigger than words. Other ontologies focus on the word meaning (e.g., Cyc, WordNet). Moreover, in some of the ontologies there is one-to-one mapping between concepts and words in a natural language, while in others many concepts may not map to any word in a language or may map to more than one in the same language.

There are also differences with respect to their degree of language dependency. Some linguistic ontologies depend totally on a single language (e.g., WordNet), others are valid for several languages (e.g., GUM), some others contain a language-dependent part and a language-independent part (e.g., EuroWordNet), and others are language independent (e.g., Mikrokosmos). The origin and motivation of these ontologies are varied: online lexical databases (e.g., WordNet), ontologies for machine translation (e.g., Sensus), and ontologies for natural language generation (e.g., GUM), etc.

In the following subsections, some of the most relevant linguistic ontologies that have been applied in natural language applications are described.

3.2.1 Cyc Project

The Cyc project (Lenat & Guha, 1990) is an artificial intelligence project whose primary goal was to build a large knowledge base containing a store of formalized background knowledge suitable for supporting human-like reasoning and problem-solving tasks in a variety of domains. The objective was to codify, in machine-usable form, millions of pieces of knowledge that compose human common sense.

Currently, the Cyc KB contains more than 2.2 million assertions (facts and rules) describing more than 230,000 terms, including nearly 15,000 predicates.

Although CYC was originally motivated by the need for knowledge systems to have world knowledge it has been tested in natural-language applications more than in knowledge-systems applications. The Cyc Knowledge Base is subdivided into three layers, the upper, the middle, and the lower ontologies. Each ontology contains information with different levels of generality:

- **Upper ontology:** contains abstract or highly structural concepts. This ontology is the smallest but most broadly referenced one of the Cyc ontology. It is intended to capture concepts such as temporality, mathematics, and relationship types. It allows for the representation of individuals and their relation to space, time, and human perception.
- **Middle ontology:** captures a layer of abstraction that is widely used, but not universal to all knowledge engineering efforts. Some examples might be the geospatial relationships, broad knowledge of human interactions, or everyday items and events.

- **Lower ontology:** contains domain-specific "leaf level" knowledge, such as that specific to a field of study like medicine, chemistry, or information about a particular person or nation. This layer accounts for the largest segment of knowledge in the KB, but the least broadly applicable.

OpenCyc (Cycorp, 2013) is a freely available subset of the knowledge base. The last release of OpenCyc contains a subset of Cyc with more than 230,000 terms. This knowledge base also includes an executable Knowledge Server that includes an inference engine and other tools for accessing, utilizing, and extending the content of the knowledge base. OpenCyc provides a foundation of ontological concepts that can be immediately used and easily extended. CycL is an expressive language that supports the OpenCyc ontology. The definitional vocabulary used to express the taxonomic information in OpenCyc reflects many of the expression needs and issues encountered over the course of building the Cyc Knowledge Base.

With regard to the applications of the OpenCyc knowledge base, it can be used as the basis of a wide variety of intelligent applications such as semantic data integration, rich domain modelling, text understanding, domain-specific expert systems and game AIs.

ResearchCyc represents the Cyc software and knowledge base at no cost to the research community under a ResearchCyc license. The current release (1.1) contains the complete non-proprietary content of the Cyc knowledge for research-only purposes.

The knowledge in OpenCyc is a small subset of the knowledge in the ResearchCyc KB. However, since the bulk of the knowledge is definitional, that subset represents a large body of background knowledge and provides the key mechanism that enables formal knowledge to be shared among users of the ontology. OpenCyc has been used to support research in areas ranging from automatic pruning of irrelevant knowledge (Conesa & Olivé, 2004), to the integration of Semantic Web metadata (Sicilia, García, Sánchez, & Rodríguez, 2004), to obtaining part of speech information from a body of natural language data using machine learning (O'Hara, et al., 2003), (Matuszek, et al., 2005).

The OpenCyc system has been criticized for being restrictive, particularly with respect to instance-level knowledge, for lacking any reasoning capability, and being formally inconsistent. While the last is difficult to overcome in a very large, largely hand-tooled knowledge base, concerns about inadequate coverage of some types of knowledge have been at least partially addressed by ResearchCyc.

For commercial applications, EnterpriseCyc provides a supported version of the knowledge base and reasoning technology that includes enterprise-grade development, deployment, and administration capabilities.

3.2.2 WordNet

WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) is a very large lexical database that emerges as a proposal for a more effective combination of traditional lexicographic information and modern high-speed computation. The frequent objection to standard alphabetical procedures as dictionaries is that finding words on an alphabetical list can be tedious and time-consuming. Many people who would like to refer to a dictionary decide not to bother with it because finding the information would interrupt their work and break their train of thought.

The initial idea of WordNet was to provide a tool to use in searching dictionaries conceptually, rather than merely alphabetically. It was thought to be used in close conjunction with an online dictionary of the conventional type. The most obvious difference between WordNet and a standard dictionary is that WordNet divides the lexicon into five categories: nouns, verbs, adjectives, adverbs, and function words. Actually, WordNet contains only nouns, verbs, adjectives, and adverbs; it does not contain the function words. The realization that syntactic categories differ in subjective organization emerged first from studies of word associations. For this reason, the small set of English function words is omitted on the assumption that they are probably stored separately as part of the syntactic component of language (Garrett, 1982).

WordNet could represent a thesaurus more than a dictionary, but it is more than that. Some thesauri as Laurence Urdang's revision of Rodale's *The Synonym Finder* (Laroche, Urdang, & Rodale, 1986) and Robert L. Chapman's revision of *Roget's International Thesaurus* (Chapman, 1992) were two helpful tools in the creation of WordNet. However, the structure of WordNet has several differences with them. On one hand, alphabetical thesauri need redundant entries to register synonymy. For example, if two terms (t_1, t_2) are synonyms, the pair should be entered twice, once alphabetized under t_1 and again alphabetized under t_2. On the other hand, in topical thesauri two look-ups are required, first on an alphabetical list and again in the thesaurus proper, thus doubling a user's search time. Furthermore, WordNet includes groups words together based on their meanings and interlinks specific senses of words and labels the semantic relations among words.

An important point to understand the structure of WordNet is to know its basic design. Table 1 represents the lexical matrix that WordNet uses to organize the knowledge. In this table, word forms (terms) are listed as headings for the columns and the word meanings as headings for the rows. An entry in a cell of the matrix implies that the form in that column can be used, in an appropriate context, to express the meaning in that row. Thus, entry $E_{1,1}$ implies that word form F_1 can be used to express word meaning M_1.

Mappings between forms and meanings are many-to-many, i.e., some forms have several different meanings, and some meanings can be expressed by several different forms. On one hand, if there are two entries in the same column, the word form is polysemous. On the other hand, if there are two entries in the same row, the two word forms are synonyms, relative to a context.

Word Meanings	Word Forms				
	F_1	F_2	F_3	...	F_n
M_1	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	...	$E_{1,n}$
M_2	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$	...	$E_{2,n}$
...	...	...	...	...	
M_m	$E_{m,1}$	$E_{m,2}$	$E_{m,3}$	...	$E_{m,n}$

Table 1 - Lexical Matrix

WordNet groups nouns, verbs, adjectives and adverbs into synsets (sets of synonymous words). Each synset express a different concept and are interlinked by means of semantic and lexical relations. WordNet contains about 117,000 synsets linked to other synsets by means of a small number of conceptual relations.

3.2.2.1 Conceptual relations in WordNet

The main conceptual relation among words in WordNet is synonymy. Two expressions are considered to be synonymous if the substitution of one for the other never changes the truth value of a sentence in which the substitution is made (Ishiguro, 1990). Taking into account this definition, true synonyms are rare. For this reason, usually a weakened version of the Leibniz definition is considered: two expressions are synonymous in a linguistic context if the substitution of one for the other in that context does not alter the truth value. For instance, the first sentence shown in Table 2 if we choose bright as the target word, a suitable substitute could be brilliant, which would both maintain the meaning of the target word and at the same time fit the context.

The sun was ***bright***
The sun was ***brilliant***
His feature ***film*** debut won awards
His feature ***movie*** debut won awards
The market is ***tight*** right now
The market is ***pressured*** right now

Table 2: Example of synonyms in the same context

Another familiar relation between concepts in WordNet is antonymy. The definition of this relation is quite complex. It is often said that the antonym of a word 'x' is not-'x', but this is not always correct. For example, *rich* and *poor* are antonyms, but to say that someone is not rich does not imply that they must be poor. This relation seems to be a simple symmetric relation but is actually quite difficult, yet English speakers have little difficulty recognizing antonyms when they see them. Antonymy provides a central organizing principle for the adjectives and adverbs in WordNet. Adjectives are divided into two major classes: descriptive and relational. In one hand, descriptive adjectives are organized in terms of binary oppositions (antonymy) and similarity of meaning (synonymy). Descriptive adjectives that do not have direct antonyms are said to have indirect antonyms by virtue of their semantic similarity to adjectives that do have direct antonyms. For example, the antonym of 'heavy' is 'light', which expresses a value at the opposite pole of the 'weight' attribute. In the other hand, relational adjectives are variants of modifying nouns and are cross-referenced to the nouns contained in the WordNet database.

(Gross, Fischer, & Miller, 1989) proposed that adjective synsets should be regarded as clusters of adjectives associated by semantic similarity to a focal adjective that relates the cluster to a contrasting cluster at the opposite pole of the attribute. Thus, 'ponderous' is similar to 'heavy' and 'heavy' is the antonym of 'light', so a conceptual opposition of ponderous/light is mediated by heavy. Gross, Fischer, and Miller distinguish direct antonyms like heavy/light, which are conceptual opposites that are also lexical pairs, from indirect antonyms, like heavy/weightless, which are conceptual opposites that are not lexically paired. Under this formulation, all descriptive adjectives have antonyms; those lacking direct antonyms have indirect antonyms, i.e., are synonyms of adjectives that have direct antonyms.

An example of this network of antonym/synonymy is illustrated in Figure 6 for the cluster of adjectives around the direct antonyms, 'wet/dry'. For example, moist does not have a direct antonym, but its indirect antonym can be found via the path, moist ->wet ->dry.

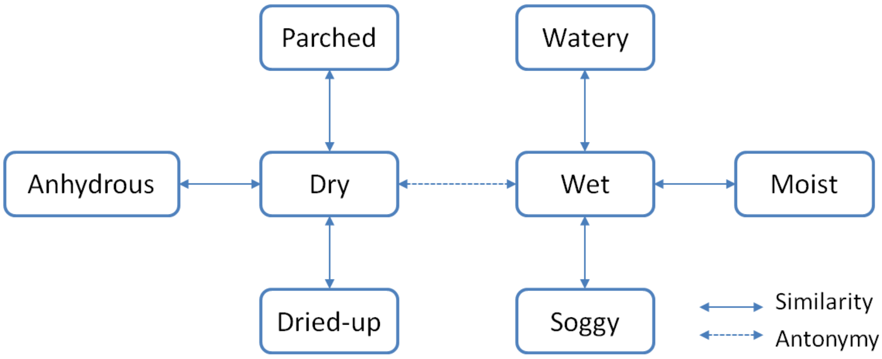

Figure 6 - Indirect Antonym Example

In the other hand, the other type of adjectives represented in WordNet is the relational ones. These adjectives differ from the descriptive ones in that they do not relate to an attribute. They mean something like 'of', 'relating/pertaining to', or 'associated with' some noun, i.e., they play a role similar to that of a modifying noun. Examples of these adjectives are 'dental' in 'dental hygiene' that refers to 'tooth' and 'chemical' in 'chemical engineer' that refers to 'chemistry'.

Hyponymy (*is-a* relation), also covered by WordNet, is the semantic relationship between a specific word and a general word when the former is included within the latter. For example, 'maple' is a hyponym of 'tree', and 'tree' is a hyponym of 'plant'. This relation is transitive and asymmetrical, and there is normally a single superordinate, it generates a hierarchical semantic structure, in which a hyponym is said to be below its superordinate. Such hierarchical representations are widely used in the construction of information retrieval systems, where they are called inheritance systems (Touretzky, 1986): a hyponym inherits all the features of the more generic concept and adds at least one feature that distinguishes it from its superordinate and from any other hyponyms of that superordinate. Figure 7 represents an example of hypernymy for the synset 'car'.

Another semantic relationship in WordNet is the meronymy (Cruse, 1986), also known as part-whole (or has-a) relation. This relation has an inverse, i.e if W_1 is a meronym of W_2, then W_2 is said to be a holonym of W_1. The part-of relation is often compared to the *is-a* relation because both are asymmetric, transitive, and can relate terms hierarchically. That is to say, parts can have parts: 'finger' is a part of 'hand', 'hand' is a part of 'arm', 'arm' is a part of 'body', so the term 'finger' is a meronym of the term 'hand', 'hand is a meronym of 'arm' and, finally, 'arm' is a meronym of 'body'. Figure 8 represents an example of holonymy for the synset 'car'.

Verbs in WordNet are organized in different files, apart from the nouns, adjectives and adverbs. They also have different relationships. These differences are due to the higher polysemy of them that suggest that verb meanings are more flexible than noun meanings. In fact, verbs (more frequently in English) can change their meanings depending on the kinds of noun arguments with which they co-occur, whereas the meanings of nouns tend to be more stable in the presence of different verbs.

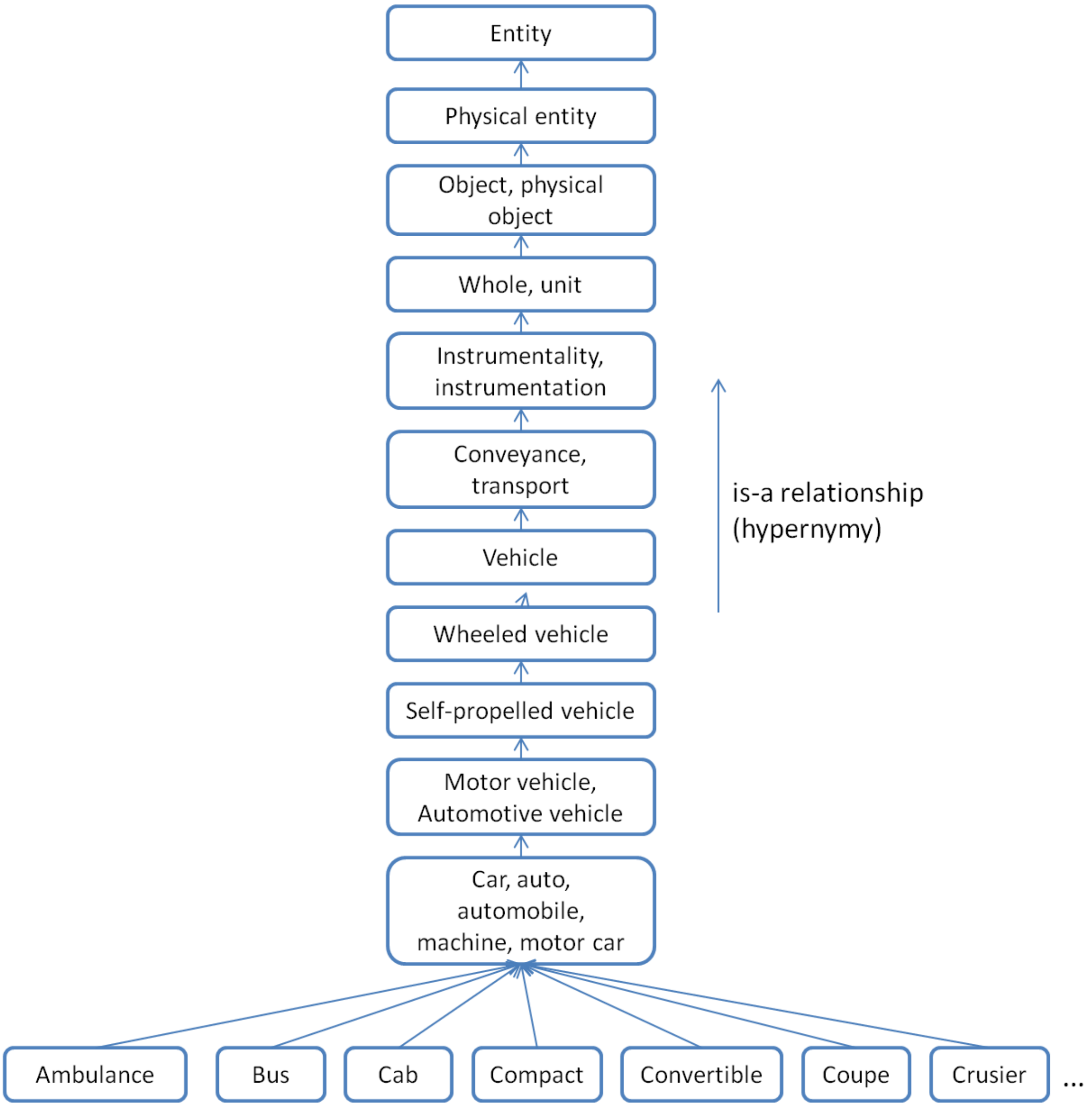

Figure 7 – Example of hypernymy: synset“car”

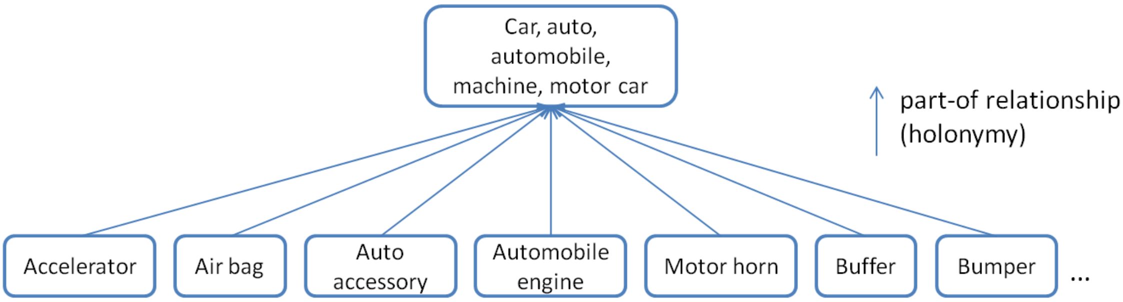

Figure 8 - Example of holonymy: synset“car”

Verbs are divided into 15 files, largely on the basis of semantic criteria. All but one of these files correspond to what linguists have called semantic domains: verbs of bodily care and functions, change, cognition, communication, competition, consumption, contact, creation, emotion, motion, perception, possession, social interaction, and weather verbs. Virtually all the verbs in these files denote events or actions. Another file contains verbs referring to states, such as 'suffice', 'belong', and 'resemble' that could not be integrated into the other files.

Lexical entailment is a kind of unilateral relationship covered by WordNet. It can be defined in this way: if a verb V1 entails a verb V2, then V2 necessarily holds if the first one does. For example, 'snore' lexically entails 'sleep' because a sentence like 'He is snoring' entails 'He is sleeping', so 'snore' necessarily implies 'sleep'. It can be said that the entailment relation between verbs is similar to meronymy between nouns.

A particular kind of lexical entailment is the troponymy. In this relation every verb V_1 that is a troponym of a more general V_2 verb also entails V_2. For example, in the verbs 'limp' and 'walk' are related by troponymy because 'to limp' is also 'to walk' in a certain manner. In addition, those verbs are also in an entailment relationship; in the sentences 'she is limping' and 'she is walking', walking can be said to be a part of limping.

In WordNet, each kind of verb is organized in a different structure. Some of them are taxonomically distributed by means of the troponymy relation (verbs of creation, competition, communication, contact, motion and consumption). Other verbs, such as the stative verbs and verbs of change have an entirely different structure. They are organized in terms of synonymy and opposition.

The semantic organization of adverbs in WordNet is simple: they are maintained in a single file and there is not a hierarchical or cluster structure as occurs for nouns or adjectives. So far, adverbs have not been categorized and the majority of them are derived from adjectives via morphological affixation.

WordNet has been used for a number of different purposes in information systems, including word sense disambiguation, information retrieval, automatic text classification, automatic text summarization and machine translation (Morato, Marzal, Lloréns, & Moreiro, 2004). WordNet has an essential classification of linguistic nature which distributes the concepts in just four categories

3.2.3 EuroWordNet

EuroWordNet (Vossen, 1998) is a multilingual database with wordnets in several European languages, (Dutch, Italian, Spanish, German, French, Czech and Estonian) structured along the same lines as the Princeton WordNet and specifically the version 1.5. Each European wordnet is stored in a central lexical database and each concept is linked to the closest synset in the Princeton WordNet 1.5. In the EuroWordNet database it is possible to go from one concept in a wordnet to a concept in another wordnet, which is linked to the same WordNet 1.5 concept. Synsets linked to the sameWordNet1.5 synset are supposed to be equivalent or close in meaning and can then be compared.

The EuroWordNet database maintains the notions of synset and the main relations from WordNet (hypernymy, hyponymy, etc). Nevertheless, some specific changes have been made in the multilingual database design to attempt the following features:

- To maintain language-specific relations in the wordnets.
- To achieve the maximal compatibility across the different wordnets.
- To build the wordnets independently and reusing existing resources.

To maintain the language-specific structures in the wordnets, a distinction is made between the language-specific modules and a separate language-independent module.

Synsets in the different languages are related to the WordNet 5.1 synsets through the Inter-Lingual-Index (ILI). Each synset in the monolingual wordnets has at least one equivalence relation with a record in the ILI. The language-specific synsets that are linked to the same ILI record, are equivalent across the languages. In the Figure 9 an example for the concept 'drive' is presented for the English and Spanish wordnets.

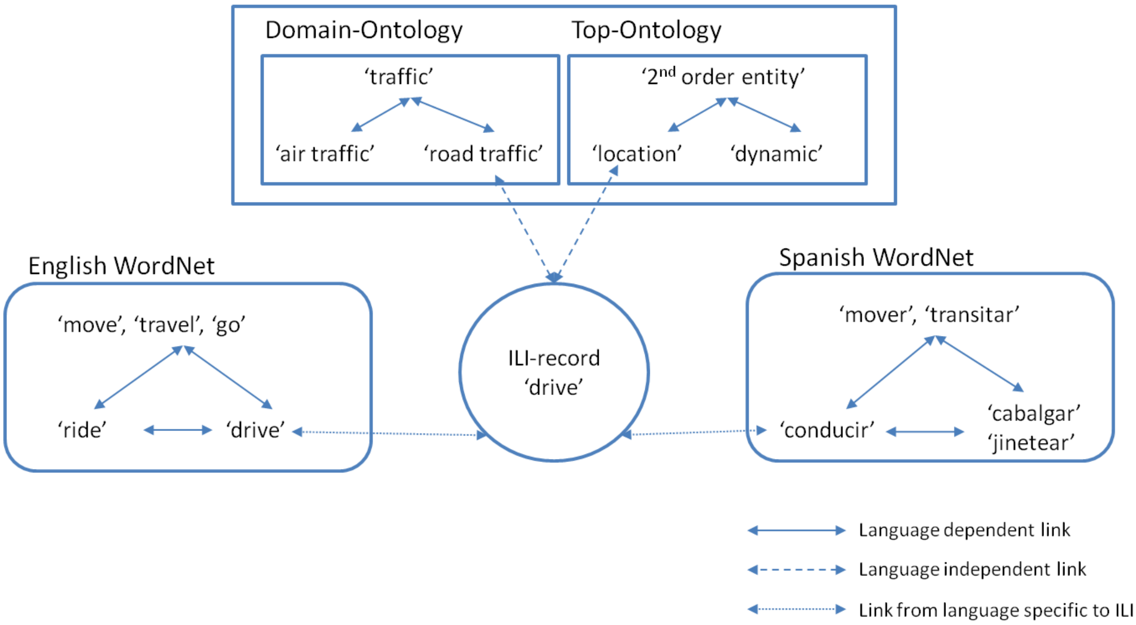

Figure 9 - ILI representation example

The objective of the Top Ontology is to provide a common framework for the most important concepts in all the wordnets and its classification has been verified by different sites, so that it holds for all the language-specific wordnets. The modular database design shown in Figure 9 has several advantages:

- It is possible to expand words in one language to related words in another language via the ILI.
- Language-dependent differences can be maintained in the individual wordnets without affecting the other wordnets.
- Different wordnets can be developed at different sites independently.
- Top ontology concepts and domain knowledge can be stored once and can be available to all the languages-specific modules.

The ability of giving information on the lexical patterns across languages is very useful for machine translation and language learning systems. In addition, EuroWordNet database can be used for the monolingual and cross-lingual information retrieval, question/answering systems, language understanding and expert systems.

3.2.4 Global WordNet Association

The Global WordNet Association (Global WordNet Association, 2000) is a free, public and non-commercial organization that provides a platform for maintaining, standardizing and interlinking wordnets for all languages in the world. GWA involves the coordination and integration of the current initiatives: the Princeton WordNet and EuroWordNet.

Among the main objectives, the GWA aims to standardize the specification of lexical semantic relations, the notion of a synset, the degrees of polysemy and the Inter-Lingual-Index for interlinking the wordnets of different languages. In addition, GWA is working on the developing of a common XML representation for wordnet data and preparing sense-tagged corpora in all the linked languages. Furthermore, the association aims to include guidelines and methodologies for building wordnets in new languages, consistency checking and evaluation modules.

In the Table 18 located in the Appendix is shown a table that contains the Wordnets in the World that are part of the Global WordNet Association.

3.2.5 Projects derived from WordNet

In this subsection, will be reviewed some of the most important related projects derived from WordNet. One of them is BabelNet (Navigli & Pontezzo, 2012), which is a very large multilingual encyclopedic dictionary and a semantic network which connects concepts and named entities in a network of semantic relations between concepts called 'Babel synsets'. Each Babel synset represents a given meaning and contains all the terms which express that meaning (synonyms) in a range of different languages.

The current version of the project, BabelNet 2.0 contains concepts from 50 languages and is built from the following resources:

- WordNet 3.0, the last release of the English computational lexicon.
- Open Multilingual WordNet that includes 22 wordnets from different languages.
- Wikipedia, the largest collaborative multilingual Web encyclopedia.
- OmegaWiki, a large collaborative multilingual dictionary.

Another important project which is derived from WordNet is The Suggested Upper Merged Ontology (SUMO) (Pease, Niles, & Li, 2002). SUMO consists of a formal ontology and a set of domain ontologies and was built by merging publicly available ontological content into a single, comprehensive, and cohesive structure. It also has been mapped to the entire WordNet lexicon. Currently, SUMO is the largest public and formal ontology available. It covers about 25,000 terms and 80,000 axioms (if the domain ontologies are included). Domain ontologies are relative to different areas: communication, countries, distributed computing, economy, automobiles, food, sports, shopping catalogues, music, etc.

YAGO (Suchanek, Kasneci, & Weikum, 2007) is also a semantic knowledge database that currently includes information of more than ten million entities and contains more than 120 million facts about them. The knowledge have been extracted from Wikipedia and unified with WordNet. In addition to the hierarchical relationship, YAGO includes a temporal and a spatial dimension to many of its facts and entities. With regard to the domains, YAGO has some thematic ones such as "science" or "music" extracted from the WordNet domain knowledge.

3.3 Methodologies, Languages and Tools for Building Ontologies

During the last two decades, ontology development has become an engineering discipline, the Ontology Engineering, which refers to the set of activities focused on the ontology development process and the ontological life cycle. When an ontology are going to be built, several issues need to be considered:

- The set of existing methods or methodologies for building ontologies. It is convenient to choose the appropriate method depending on the need of the ontology to be developed.
- The language(s) to be used to implement the ontology, taken into account its expressiveness, and if it is necessary, its ability to integrate the ontology in an application and/or if it is appropriate for exchanging information between different applications.
- The tool(s) that will give support to the ontology development process, taking into account the features needed, for example, if the existence of an inference engine.

In the following subsection, some methodologies commonly used for building ontologies will be summarized. In Section 3.3.2, the most used ontology languages will be described and, finally, some tools usually used during the ontology development process are presented in Section 3.3.3.

3.3.1 Methodological Tools

Since ontology development is not an easy task, some methodologies have been emerging during the last decades. In the following sub-sections, a brief overview of the most referred methodologies is presented from oldest to newest.

3.3.1.1 METHONTOLOGY

METHONTOLOGY (Fernández, Gómez-Pérez, & Juristo, 1997) has been developed at Polytechnic University of Madrid and is based on IEEE standards for Developing Software Life Cycle Processes. This methodology provides the guidelines for the following purposes:

- Project management process: guidelines for the planning, the project control and the quality control.
- Ontology development process: guidelines for envisioned use of the ontology, conceptualization of the target domain, the formalization and implementation of the ontology, etc.
- Support activities: guidelines for knowledge acquisition, evaluation, ontology integration, documentation, etc.

3.3.1.2 On-To-Knowledge

On-To-Knowledge (Sure, et al., 2003) is a methodology developed at Karlsruhe University and it is based on a two-loop architecture: knowledge process and knowledge meta process for introducing and maintaining ontology-based knowledge management.

On one hand, the knowledge process is a normal knowledge use and evolution process. On the other hand, the knowledge meta process is a methodology of ontology development composed of five mayor steps: 1) a feasibility study, 2) the kick off where ontology requirements are identified, 3) the refinement where a mature and application-oriented ontology is produced, 4) evaluation and 5) maintenance.

3.3.1.3 The DILIGENT Methodology

The DILIGENT methodology (Pinto, Staab, & Tempich, 2004) is a methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. This methodology was developed to support domain experts in a distributed setting to engineer and evolve ontologies. It is focused on collaborative and distributed ontology engineering. The ontology development process includes the following five activities: 1) building, 2) local adaption, 3) analysis, 4) revision and 5) local update.

3.3.1.4 NeOn Methodology

The NeOn Methodology (Suárez-Figueroa, Gómez-Pérez, & Fernández-López, 2012) for building ontology networks is a scenario-based methodology that supports a knowledge reuse approach, collaborative aspects of ontology development and dynamic evolution of ontology in distributed environments.

The main features of the NeOn methodology are:

- A set of scenarios for building ontologies and ontology networks, emphasizing the reusability, the collaboration and the dynamism.
- The NeOn Glossary which identifies processes and activities carried out when ontology networks are collaboratively build by different teams.
- Methodological guidelines which includes the ontology requirements specification, the ontology localization, the scheduling, etc.

3.3.2 Major Ontology Representation Languages

One of the key decisions to take into account in the ontology development process is to select the language (or set of languages) in which the ontology will be implemented. In the last decades, many ontology implementation languages have been created and other general knowledge representation languages and systems have been used for implementing ontologies.

In the following subsections, an overview of the major ontology languages developed in the scope of the W3C Semantic Web Activity (MIT, ERCIM, Keio, Beihang, 2013) is presented.

3.3.2.1 First Ontology Markup Languages

The first ontology markup language to appear was SHOE (Luke & Heflin, 2000). SHOE is a language that combines frames and rules and was built as an extension of HTML, in 1996. It used tags different from those of the HTML specification, thus allowing the insertion of ontologies in HTML documents. Later its syntax was adapted to XML.

The majority of ontology markup languages developed following SHOE are based on XML. XOL (Karp, Chaudhri, & Thomere, 1999) was developed as a XML adaptation of a small subset of primitives from the OKBC protocol (Chaudhri, Farquhar, Fikes, Karp, & Rice, 1998).

3.3.2.2 RDF and RDF Schema

RDF (Lassila, Swick, Wide, & Consortium, 1998) stands for Resource Description Framework. This language was developed by the W3C (Berners-Lee, World Wide Web Consortium, 1994) as a semantic-network based language to describe Web resources. Its data model is equivalent to the semantic networks formalism, consisting of three object types: resources, properties and statements.

The RDF data model does not have mechanism for defining the relationships between properties and resources. This is the role of the RDF Vocabulary Description language, also known as RDF Schema (Brickley & Guha, 2004) language was also proposed by the W3C as an extension to RDF with frame-based primitives. The combination of both RDF and RDF Schema is normally known as RDF(S). It employs the triplet model <object, attribute, value>, in which object is called resource representing a web page. A triplet itself can be an object and a value. Value can take a string or resource. Object and value are considered as a node and attribute as a link between nodes. These languages have established the foundations of the Semantic Web (Berners-Lee & Fischetti, Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web, 1999).

3.3.2.3 OWL

The Web Ontology Language (OWL) (Dean & Schreiber, 2004) is the result of the work of the W3C Web Ontology Working Group. It is derived from the DAML+OIL language which was created in the context of the DARPA project DAML (Pagels, 2000) and it is based on two languages: the previous DAML-ONT specification and the OIL language (Horrocks, et al., 2000).

OWL is built upon RDF(S), has a layered structure and is divided into three sublanguages: OWL Lite, OWL DL and OWL Full. This ontology language is grounded on Description Logics (Baader, Lutz, Sturm, & Wolter, 2002) and its semantics is described in two different ways: as an extension of the RDF(S) model theory and as a direct model-theoretic semantics of OWL. Its design principle includes developing a standard language for ontology representation to enable semantic web, and hence extensibility, modifiability and interoperability are given the highest priority. At the same time, it tries to achieve a good trade-off between scalability and expressive power.

OWL 2 (Motik, Cuenca Grau, Horrocks, Wu, Fokoue, & Lutz, 2009) is an extension and revision of OWL that adds new functionality with respect to OWL. This language includes three different profiles that offer important advantages in particular scenarios depending on the ontological needs, for example, the size of the ontology or the type of access to the data.

3.3.3 Leading Ontology Tools

The large number and variety of tools used to manage and exploit ontologies makes it necessary to classify them according to their functionalities. They provide interfaces that help users carry out some of the main activities of the ontology development process, conceptualization, implementation, consistency checking, and documentation. Semantic technologies can be categorized into five different dimensions based in the Semantic Web Framework (García-Castro, Gómez-Pérez, & Muñoz-García, 2008). These dimensions are and are described below:

- **Ontology Management:** This dimension includes components that manage ontology-related information.
- **Querying and reasoning:** This dimension includes components that generate and process queries.

- **Ontology engineering:** This dimension includes components that provide functionalities to develop and manage ontologies.
- **Ontology processing:** This dimension includes components that process ontologies.
- **Instance generation:** This dimension includes components that generate ontology instances.

Each dimension has several components that are described in the Appendix in Table 19. In addition, some of the relevant tools for each component are listed in the same table.

Regarding the ontology engineering dimensions, there are software platforms that cover more than one of the mentioned components and are able to support most of the activities in the ontology development process. Some examples of these ontology environments widely used are Protégé (Informatics, 2013), the TopBraid Composer (TopQuadrant, 2013) and the NeOn Toolkit (NeOn Foundation, 2013).

3.3.3.1 Protégé

Protégé (Musen, 2015) is a free, open source and popular ontology editor and a knowledge-based framework that supports modelling ontologies via a web client or a desktop client. Protégé ontologies can be developed in a variety of formats including OWL, RDF(S) and XML Schema. The main Protégé functions are:

- Load and save OWL and RDF ontologies.
- Edit and visualize classes, properties and rules.
- Define logical class characteristics as OWL expressions.
- Execute reasoners such as description logic classifiers.
- Edit OWL individuals for Semantic Web markup.

This tool is supported by a strong community of developers in areas as diverse as biomedicine, intelligence gathering, and corporate modelling.

3.3.3.2 TopBraid Composer

Another important tool to modelling semantic applications is the TopBraid Composer (TopQuadrant, 2001). It is fully compliant with W3C standards and offers complete support for developing, managing and testing configurations of ontologies and linked data.

TopBraid Composer incorporates a flexible and extensible framework with a published API for developing semantic client/server or browser-based solutions that can integrate different applications and data sources.

3.3.3.3 NeOn Toolkit

Other example is the NeOn Toolkit (NeOn Foundation, 2013). It is an ontology engineering environment originally developed as part of the NeOn Project and it covers a variety of ontology engineering activities, including annotation and documentation development, knowledge acquisition management and ontology matching. It has an open and modular architecture, which the NeOn Toolkit inherits from its underlying platform, Eclipse (The Eclipse Foundation, 2015). Eclipse is a very rich development environment, which is widely adopted in the programming world and which perfectly fits to the modelling paradigm for ontologies.

3.4 Semantic Similarity Measures

In this section, two main objectives will be exposed in relation to the semantic similarity measures applied to ontologies. First, it will aim to provide an overview of the different types of approaches available for the comparison of concepts in ontologies and, in so doing, to identify the foundations on which the desired similarity measure may be modelled. Secondly, it aims to select the best way to evaluate the results yielded from this desired similarity measure according to other studies regarding similarity metrics assessment.

3.4.1 Types of methods

The existing methods for the calculation of the semantic similarity measure between terms can be classified into five categories of methods: based on semantic distance, based on information content, based on properties of terms, based on ontology hierarchy and hybrid methods.

Since each category of methods has its own traits, it is important to know which method is suitable for the application of interest. In the following subsections, the characteristics of each category will be summarized, pointing out main characteristics, the advantages and disadvantages of these methods.

3.4.1.1 Methods based on semantic distance.

This approach calculates the similarity between terms calculating the distance between the concepts corresponding to these terms in an ontology with hierarchical structure. This method is the simplest and most intuitive similarity measure. It is based mainly on the counting of the number of edges in a path between two terms on a graph (Rada, Mili, Bicknell, & Blettner, 1989).

The sorter the distance between the two terms in the path, the higher the similarity between them. If there are multiple paths between the nodes, the shortest or the average distance of all paths may be used. Some factors are usually considered in this type of methods:

- **Depth of nodes:** the deeper the nodes located in, the higher the difference between the nodes.
- **Density in the ontology graph:** the higher the density of the nodes in the graph, the nearer the distance between nodes.
- **Types of links:** most common relation is that of type *is-a*, but other relations such as part-of or substance-of are associated with the weight of the edges between nodes.
- **Weights of links:** the similarity between terms is affected by the edges that connect the nodes can vary depending on different semantic weights.

Within this group of methods, some representative algorithms are the shortest path proposed by (Rada, Mili, Bicknell, & Blettner, 1989), the connection weight method (Sussna, 1993) and the common path method (Wu & Palmer, 1994).

The method proposed by (Rada, Mili, Bicknell, & Blettner, 1989) suggests the shortest path between two nodes was the simplest approach for measuring the distance between two terms. This distance is calculated with the formula:

$$Sim(C1, C2) = 2MAX - L$$

where C1 and C2 are concepts in the hierarchical graph, MAX is the maximum path on the hierarchy, and L the shortest path between the compared concepts. The main advantage of this method is the low complexity in calculation.

Alternatively, (Sussna, 1993) employed the notion of semantic distance between network nodes in order to improve the results in the word sense disambiguation area. Input text terms with multiple senses were disambiguated by finding the combination of senses from a set of contiguous terms which minimizes total pairwise distance between senses. Sussna proposed an edge weight measure which considered the density of the graph, the depths of nodes, and the types of connections. The distance between adjacent concepts C1 and C2 was defined as

$$w(C1, C2) = \frac{\mathrm{w}\left(\mathrm{C1} \overset{\mathrm{r}}{\rightarrow} \mathrm{C2}\right) + w\left(\mathrm{C1} \overset{\mathrm{r'}}{\rightarrow} \mathrm{C2}\right)}{2d},$$

$$given\; w\left(C1 \overset{r}{\rightarrow} C2\right) = max_r - \frac{max_r - min_r}{n_r(x)}$$

where $\overset{r}{\rightarrow}$ is a relation of type r, $\overset{r'}{\rightarrow}$ its inverse, d the depth of the deeper node, max_r and $= min_r$ the maximum and minimum weights for a relation of type *r*, respectively, and $n_r(x)$ the number of relations of type *r* leaving node *x.* This method achieved an improvement in reducing the ambiguousness of multiple sense words by discovering the combination of senses from a set of common terms that minimizes the total pairwise distance between senses.

Another semantic distance-based method is the common path technique proposed by (Wu & Palmer, 1994). This method calculated the similarity directly by the length of the path from the lowest common ancestor (LCA) of the two concepts to the root node. The calculation of the semantic distance between two concepts C1 and C2 was denoted as

$$Sim(C1, C2) = \frac{2H}{D_1 + D_2 + 2H}$$

where *D1* and *D2*, are, respectively, the shortest paths from *C1* and *C2* to their LCA, and *H* the shortest path from the LCA to the root.

3.4.1.2 Methods based on information content.

This type of methods determines the semantic similarity between two concepts based on Information Content (IC) of their lowest common ancestor (LCA) node. The IC gives a measure of how specific an informative a concept is. The IC of a concept *c* is calculated as the negative log likeness

$$\mathrm{IC}(c) = -\log \mathrm{P}(c)$$

Where $\mathrm{P}(c)$ is the probability of occurrence of the concept c in a specific corpus. On the ontology hierarchy, the occurrence probability of a concept decreases when the layer of the concept goes deeper, and the IC of the concept increases. Therefore, the lower a concept in the hierarchy, the greater its IC.

Some representative researches belonging to this category are the proposed by (Resnik, 1999).

Resnik's method (Resnik, 1999) used a taxonomy with multiple inheritance with a semantic similarity measure of terms based on the notion of information content. This measure defined the IC of a concept C as the negative algorithm of the probability of its occurrence. The similarity between two concepts C1 and C2 was defined by the maximal IC of all concepts subsuming both C1 and C2 and its calculated as follows:

$$Sim(C1, C2) = \max_{c \in S(C1,C2)} [-\log P(C)]$$

where S(C1,C2) is defined as the set of all the parents for both concepts C1 and C2. The main advantages of this method are the easy formulation and simple calculation. However, this method is only suitable for the ontology hierarchy with simple relations, so it cannot be applied to the concepts with either part-of relations or inferior relations.

Lin's research proposed an alternative information theoretic approach. In addition to the parent commonality of the query concepts, this method took into account the IC associated with the query concepts. Three basic assumptions were given by Lin in calculating the similarity between two concepts. They are defined as follows:

- The similarity between two concepts is associated with their common properties: the more the common properties, the higher their similarity.
- The similarity between two concepts is associated with their difference: the more the difference, the lower their similarity.
- The similarity between two terms reaches the maximum value when they are totally the same.

The similarity measure between two concepts C1 and C2 is based on these assumptions and is defined as follows:

$$Sim(C1, C2) = \frac{2logP(C_0)}{logP(C1) + \log P(C2)}$$

where C_0 is the LCA of the two concepts (C1, C2) and P(C1) and P(C2) are the probabilities of ocurrence of both concepts. This method can be seen as a normalized version of the Resnik's one.

Another similarity measure based on the IC was proposed by Jiang and Conrath (Jiang & Conrath, 1997). This approach combined the edge-based method of the edge counting scheme with the node-based method of the information content calculation. This measure takes into account the depth of nodes, the density around the nodes and the type of connections. Jiang and Conrath's similarity measure is defined as follows:

$$Sim(C1, C2) = IC(C1) + IC(C2) - 2 \times IC(LCA(C1, C2))$$

In both methods, Lin and Jiang and Conrath, are proportional to the IC differences between the concepts and their common ancestor, independently of the absolute IC of the ancestor. To overcome this issue, (Schickel-Zuber, 2007) proposed another method based on Lin's measure but using the probability of annotation of the most informative common ancestor (MICA) and the calculation is defined as follows:

$$Sim(C1, C2) = \max_{c \in S(C1,C2)} \left(\frac{2 \times \log P(C)}{\log P(C1) + \log P(C2)} \times (1 + P(C)) \right)$$

All this measures does not take into account the existence of disjoint common ancestors (DCAs). To overcome this limitation, (Couto, Silva, & Coutinho, 2007) proposed a method named GraSM, in which the IC of the MICA was replaced by the average IC of all DCA.

3.4.1.3 Methods based on properties of terms.

In this type of methods, terms are represented as collections of features and simple set of operations are applied to estimate semantic similarity between terms. Methods based on properties of terms basically consist of three components: distinct properties of concept C1 to Concept C2, distinct properties of concept C2 to concept C1, and common properties of concepts C1 and C2.

One of these methods is the proposed by (Tversky, 1977), who defined a semantic similarity measure according to the common and distinct features of terms. The calculation of the similarity measure between two concepts C1 and C2 based on Tversky's model is defined as follows:

$$Sim(C1, C2) = \frac{|D_1 + D_2|}{|D_1 \cap D_2| + \mu |D_1/D_2| + (\mu - 1)|D_2/D_1|} \quad \text{for } 0 \leq \mu \leq 1$$

where D_1 and D_2 correspond to sets of properties of the concepts C1 and C2, respectively. || determines the cardinality of a set, and μ a degree of importance of the no common properties.

Other method similar to the Tversky's one is the Matching-Distance Similarity Measure (MDSM). This model was developed in order to calculate the similarity distances of geospatial concepts.

3.4.1.4 Hybrid methods.

This type of methods usually take into account several features of the methods described above such as the ontology hierarchy, the information content, the depth of the LCA node, etc.

One of the representative methods was OSS in which a similarity measure for hierarchical ontologies called ontology structure-based similarity was defined (Schickel-Zuber, 2007). The major ingredient of this method is the computation of a-priori score of a concept *c*, *(APS(c))*, which shares some similarities with IC (i.e., both are calculated from the topology and structure of the ontology reflecting the information contained within and between the concepts).

Additionally, several hybrid methods have also been defined in an attempt to improve the results of the other techniques. In (Jiang & Conrath, 1997), for example, a combined model is defined that is derived from the edge-based notion by adding information content as a decision factor. The link strength between two concepts is defined as the difference of information content between them.

3.4.2 Similarity Measures Applications

With the aim of collecting all different methods and approaches, SimPack, a generic Java library of similarity measures for use in ontologies, has been created (Bernstein, Kaufmann, Kiefer, & Bürki, 2005) and includes the implementation of ontology-based similarity methods (including edge-based and node-based measures). It is important to note that the majority of the techniques described to define semantic similarity between concepts have been applied to hierarchical ontologies whose structure takes into account only one or two dimensions in the same graph.

For example, WordNet (Fellbaum, 1998) consists of an ontological graph with over 100,000 concepts and whose edges model the *is-a* and part-of relationships. A Perl module (Pedersen, Patwardhan, & Michelizzi, 2004) was implemented for this lexical database with a variety of semantic similarity measures. Another example of application is the Gene Ontology (Department of Genetics, Stanford University School of Medicine, 2000), one of the most important ontologies within the bioinformatics community, with over 20,000 concepts and modelling *is-a* and part-of relationships in the same graph. Thus, while none of the techniques described in this section can be supposed to be appropriate in dealing with more than two dimensions of similarity, they can nevertheless be useful to attempt to define some of the dimensions in the present study's ontological model.

3.4.3 Similarity Evaluation Techniques

The second aim of the present section is to review the evaluation techniques for ontological similarity functions used in the area. The gold standard established in the majority of the experimental evaluations of similarity (Resnik, Using information content to evaluate semantic similarity in a taxonomy, 1995) (Jiang & Conrath, 1997) (Altintas, Karsligil, & Coskun, 2005) (Schickel-Zuber, 2007) (Bernstein, Kaufmann, Kiefer, & Bürki, 2005) is based on the experiment described in (Miller & Charles, 1991) which has become the benchmark for determining the similarity of words in natural language processing research. This experiment relies on the similarity assessments made by thirty eight university students when provided with thirty name pairs chosen a priori to cover high, intermediate and low levels of similarity and when asked to assess the similarity of their meaning on a five value discrete scale from zero (no similarity) to four (perfect synonymy). The average of scored values represents a good estimation of the degree of similarity between two terms.

In certain evaluations based on human judgment (Inkpen, 2007) (Bernstein, Kaufmann, Kiefer, & Bürki, 2005), variations in the number of participants or the way to administer the questionnaire have been introduced. In one of these studies (Bernstein, Kaufmann, Kiefer, & Bürki, 2005), a website containing a survey tool was designed to perform the evaluation. In the Web experiment, subjects were asked to assess the similarity between seventy three pairs of concepts on a scale from one (no similarity) to five (identical). Finally, subjects were also given the possibility of adding comments to their assessment. To evaluate the quality of the similarity measures, its results were compared with the test subjects' assessments using the corrected Spearman rank correlation coefficient.

3.5 Automatical Learning of Ontological Knowledge

In the first place, it is necessary to define some processes that are in charge of the acquisition and maintenance of the knowledge on an ontology. The **Ontology population** is the task of adding new terms, concepts, instances of concepts and relations to the ontology. **Ontology enrichment** is the task of extending an existing ontology with additional concepts and semantic relations and placing them at the correct position in the ontology. During both processes, can appear some inconsistencies in the knowledge acquired that need to be detected and resolved. **Inconsistency resolution** is the task of resolving inconsistencies that appear in an ontology with the view to acquire a consistent sub-ontology.

3.5.1 Ontology Learning Definition

Ontology learning is the process of acquiring, constructing or integrating, an ontology automatically or semiautomatically. The acquisition of the ontological knowledge can be performed through three major approaches:

- **Population and enrichment of an existing ontology:** The ontology is constructed by scratching or by extending an existing ontology, usually based on information extracted from the content of a specific domain.
- **By specializing a generic ontology:** creating new concepts, instances or relations related to generic ones in order to adapt it to a specific domain, such as medical, architectural or pharmacological domain.
- **Integration of existing ontologies:** The ontology is created capturing commonalities among existing ontologies that convey the same or similar domains. Several methods have been proposed in the literature, such as **merging** of ontologies to create a single coherent ontology, **aligning** ontologies by establishing links between them and allowing them to reuse information from each another or **mapping** ontologies by finding correspondence among elements in the ontologies.

Acquiring domain knowledge for constructing ontologies is a process that if it is done manually is very time-consuming and error-prone. Research on automated development of ontologies from text has become increasingly important. Thus, the automatic or semi-automatic construction, enrichment and adaptation of ontologies, the so-called ontology learning task is highly desired. The process of automatic or semi-automatic construction, enrichment and adaptation of ontologies is known as ontology learning.

3.5.2 Ontological Knowledge Learning Process

In order to acquire the knowledge that conforms or extends an ontology, some basic rules are usually followed. Some of these rules are defined in the following paragraphs.

In the first place, it is necessary to extract the new terms that are not included previously in the ontology. It is very important to take into account that several terms can refer the same real concept, these terms are named synonyms and will materialize a single concept. Failure to identify which terms are synonyms may result in the introduction of redundant concepts or relations in an ontology, which in most cases is undesirable and will be necessary to perform some corrective actions to resolve them. Many approaches have been performed in order to detect and correct these issues. Some methods to identify terms, concepts and synonyms will be detailed in Section 3.5.3.

Among relations, one type is of particular importance to ontologies, namely hierarchical ones. These are the relations that realize the taxonomy backbone of an ontology, such as the subsumption relation (also referred as "*is-a*" relation in many cases). On the other hand, non-hierarchical relations are all relations that are not used in the formation of the concept hierarchy. Relevant methods for learning relations between concepts are presented in Section 3.5.4.

A significant amount of research has been performed in ontology learning, leading to a large number of proposed approaches and practical systems. A fairly complete overview of the work performed in the field is presented in (Gómez-Pérez & Manzano-Macho), as well as in (Shamsfard & Barforoush, 2003) and in (Cimiano P. , 2006).

3.5.3 Term, Concept and Synonym Identification

Term identification is the basic requisite for the ontology learning process as is stated in Section 3.5.3. A term is the realization of a recognizable entity or object (concrete or abstract) in some specific language. When this entity has a single meaning in a multimedia corpus within a specific domain is named concept.

Generally, the term identification task can be decomposed into some subtask. For example, in (Krauthammer & Nenadic, 2004) the term identification encloses three subtasks:

- **Entity/Object recognition:** This task is responsible for finding terms that reference recognizable entities within a corpus.
- **Entity/Object classification:** This task assigns a semantic category to recognized entities. This categorization is important for the task of ontology learning, as these categories are often the concepts of the thematic domain.
- **Entity/Object mapping:** This task tries to link identified objects with relevant entities in other data sources, such as object libraries, vocabularies, lexica, thesauri and databases. One of the most relevant uses of the entity mapping is the similarities recognition between terms that exist in the data sources by identifying clusters of terms that represent the same concept (synonyms).
  Synonyms are alternative realizations of entities that refer to the same (real or abstract) entity or event in a corpus that can be thought to represent the same concept or relation. A significant amount of work has been performed mainly for text corpora, by exploiting resources such as WordNet (Fellbaum, 1998).
  With the objective of collect synonyms associated with a sense (WordNet concept), standard word sense disambiguation techniques (Dagan, Glickman, & Magnini, 2005), (Yarowsky, 1992) are applied to identify the most appropriate WordNet sense of each term.
  Other approaches try to locate term synonyms through clustering, mainly based on Harris's distributional hypothesis (Harris, 1968), according to which similar terms in meaning tend to share syntactic contexts (Lin, Concept discovery from text, 2002), (Lin, Induction of semantic classes from natural language text, 2001). More recent approaches extract synonyms by applying statistical approaches over the Web (Baroni, 2004); (Buitelaar, 2005).

3.5.3.1 Term Identification Techniques

Term identification is an important task for concept discovery for ontology learning. Many approaches have been presented in the literature mainly for the processing of textual corpora and textual information extraction and retrieval. Among the most successful ones are:

- **Statistical techniques:** measure the significance of each word with respect to other words in a corpus, based on word occurrence frequencies. TF/IDF (Salton, 1975) is often employed for this task (Damerau, 1993), possibly combined with other methods, such as latent semantic indexing (Fortuna, 2005) or taking into account co-occurrence information among phrases (Frantzi, 2000).
- **Clustering techniques:** also play an important role in object identification: recognizable entities can be clustered into groups based on various similarity measures, with each cluster being a possible object (consisting of synonyms). Approaches like (Faatz, 2002) employ clustering techniques and other resources, such as the WWW to successfully extract terms. Additionally, both frequency and clustering based approaches can be substantially enhanced through the use of natural language processing techniques, such as morphological analysis, part-of-speech tagging and syntactic analysis, as terms usually are noun phrases or obey specific part-of-speech patterns (Gupta, 2002).
- **Other techniques:** Mainly use filters and heuristics. For example, Glossex (Kozakov, 2004) filters terminological candidates using lexical cohesion and a measure of domain relevance. It also uses some additional heuristics for extracting useful terms. TermExtractor (Sclano, 2007) extracts a list of "syntactically plausible" terms and uses two entropy-based measures. The first metric, called Domain Consensus, is used to select only the terms which are used consistently throughout the corpus. The second one, Domain Relevance, is used to select only the terms that are relevant to the domain of interest. Finally, extracted terms are further filtered using Lexical Cohesion, which measures the degree of association of all the words in a terminological string.

3.5.4 Semantic Relations Learning Techniques

The purpose of this section is to show a summary of the most relevant techniques that have been presented in the literature to address the task learning of semantic relations among concepts. In most cases, these techniques are divided in two categories, in those that learn **taxonomic** relations and in those that learn **non-taxonomic** relations between concepts.

On one hand, many approaches about the learning of taxonomic relations have been developed in order to organize domain concepts into taxonomies as detailed in (Cimiano P. , 2006). On the other hand and although they have received less attention, some studies focus their efforts on identifying non-taxonomic relations (Weichselbraun, 2009), (Sánchez D. M., 2008). In the following subsections the most relevant literature about each technique is reviewed and some linguistic patterns designed to represent each relation are defined.

3.5.4.1 Learning of Taxonomical Relations

Techniques for finding taxonomic (or hierarchical) relations are generally classified into three different groups of approaches: pattern-based, clustering-based and combinations of both.

- **Pattern-based techniques:** A set of lexico-syntactic patterns are defined by the user and then applied to the texts in order to obtain instances of taxonomic relations. An example of this technique is presented in (Kilgariff, 2007). Linguistic patterns have been extensively used to develop non-supervised information extraction systems and knowledge acquisition. These approaches use regular expressions that indicate a relation of interest within the text. General lexical-syntactic patterns can be designed by hand or can be learned by using a set of pre-related concepts and domain text. One of the most important successes resulting from the application of patterns is the discovery of taxonomic relationships. (Hearst M. , 1992); (Hearst M. , 1998) studied and defined a set of domain-independent patterns for hyponymy discovery which have provided the basis for further refinements and learning approaches.

- **Clustering-based techniques:** In this type of techniques, hierarchical clustering algorithms are used for finding taxonomic relations between concepts (Moreno, Riaño, Isern, Bocio, Sánchez, & Jiménez, 2004).
- **Combined approaches:** In this case, the above two techniques are combined. Firstly, lexico-syntactic patterns are applied in the text and, secondly clustering techniques are used to filter the extracted taxonomic relations. This approach is used in some studies as in (Pivk, 2007).

One of the objectives of this thesis is the learning of concepts thorough the dialogue with humans, so the design of linguistic patterns for the taxonomy construction is a relevant task. Clustering-based techniques usually need large text corpora to be applied. In the case of the learning thorough the dialogue, short phrases are used by the interlocutor and it is needed to analyse them to extract the hierarchical information increasingly. For this reason, next section will be focused on the linguistic pattern design for the hierarchical relation construction.

Taxonomy Construction based on Linguistic Patterns

In the literature, many approaches for detecting this kind of relations have been defined. (Cimiano P. H., 2004) presents three different learning paradigms based in linguistic patterns. In the first place, (Sanderson, 1999) defines an approach that relies on the document-based notion of term subsumption. Secondly, several approaches as (Bisson, Nedellec, & Canamero, 2000), (Caraballo, 1999) claim that words or terms are semantically similar to the extent to which they share similar syntactic contexts. Finally, several researchers have attempted to find taxonomic relations expressed in texts by matching certain patterns associated to the language in which documents are presented (Ahmad, 2003), (Berland M. C., 1999).

Generally, taxonomical or hierarchical ontological relationships can be defined by instance-concept links that can be extracted from text (oral or written). One way of approach this task is discovering the ontological concept of which the discovered feature is an instance. For example, ‘a sparrow is a bird’ defines a taxonomical relationship in which the first concept ‘sparrow’ is an instance of the concept ‘bird’. This link between these two concepts could be easily detected using a linguistic pattern approach.

Pattern-based approaches are heuristic methods using regular expressions that have been successfully applied in information extraction (Sánchez D. , 2010), (Sánchez D. I., 2011). The text is scanned for instances of distinguished lexical-syntactic patterns that indicate a relation of interest. This is especially useful for detecting instances of concepts that can represent *is-a* (taxonomic) relations (Hearst M. , 1992), (Sánchez D. M., 2008).

M. Hearst described a method for the automatic acquisition of the hyponymy (taxonomical or *is-a*) lexical relation from text. In this study a set of easily recognizable lexico-syntactic patterns were identified. These patterns occurred frequently and across text genre boundaries. In (Hearst M. , 1992) several patterns to detect hyponymy relationships between concepts were presented.

Hearst proposed a set of patterns to detect taxonomical (is-a) relationships. His study pointed out that only a subset of the possible instances of the hyponymy relation would appear in a particular form. For this reason, he suggested to use as many patterns as possible. Table 3 shows a list of lexico-syntactic patterns that can be used to detect the taxonomical relation. The first column indicates the pattern (NP stands for a noun phrase), the second a sentence that exemplifies that pattern and finally the concepts hypernym and hyponym that conforms the taxonomical relation.

Pattern	Example	Hyponym(NP1, NP2)
Such NP as {NP,}* {and|or} NP	Such city as Madrid	Hyponym{city, Madrid}
NP {,} such as {NP,}* {and|or} NP	Countries as Spain	Hyponym{Spain, Country}
NP {,} including {NP,}* {and|or} NP	Pets including turtles	Hyponym{turtle, pet}
NP {,} specially {NP,}* {and|or} NP	Plants specially tulips	Hyponym{tulip, plant}
NP {,} {and|or} other NP	Jeans and other clothes	Hyponym{jeans, clothe}

Table 3 - Hearst Patterns with examples

This method for identifying taxonomic relationships between concepts could be adapted to acquire other types of semantic relations. Other linguistic patterns different from taxonomic ones can be used for that purpose, detecting non-taxonomic relations (Sánchez D. M., 2008), (Sánchez D. M.-T., 2012). For example, class descriptors such as attributes, features or meronyms, who are rarely considered when developing ontologies, are data crucial for defining concepts but even WordNet only includes a reduced amount of part-of relationships.

3.5.4.2 Learning of Non-taxonomical Relations

In the same way that have been created methods for extracting hierarchical relationships between concepts, some studies have been focused on the detection of non-taxonomic relationships between concepts.

Some researchers exploit the dependencies between terms and syntactic structure in order to extract non-taxonomic relations between concepts as is presented in (Kavalec, 2004). Other studies (Ciaramita, 2005); (Schutz, 2005) extract concept pairs exploiting dependency relations and use the chi-square test to verify the statistical significance of concept co-occurrence.

In other research (Faure & Nedellec, 1998) relations extraction is considered as learning of selective restrictions for verbs. According to this method, all terms co-occurring with a verb are clustered and each of the clusters is manually labelled.

One of the most relevant non-taxonomic relations between concepts is the part-of (or part-whole) one. A part-of relationship between concepts indicates that one concept is composed of one or more parts which are themselves instances of that or another concept.

Several studies have defined different ways to extract this type of relationship between concepts in the text. The next subsections describe, respectively, some of the main studies in the part-of relation extraction area and linguistic patterns already defined to extract this relation.

Part-Of Relation Extraction

In the literature, different ways of learning part-of relationships between concepts have been described. In (Girju, 2006), this type of relations are learned from manually pretagged text and only object-object relationships are extracted. Other authors, as (Pasca, 2007), also extracted class attributes from the Web using linguistic patterns and local statistics. This method required a list of precompiled class instances to use as seeds.

More recent studies, as (Reisinger & Pasca, 2009), present a slightly supervised approach using a set of precompiled attributes as seeds for attribute discovery on web documents and query logs. A similar approach is employed in (Ravi & Pasca, 2008) but additionally relying on the HTML structure of web documents to identify relevant attributes.

As the same way that hypernymy, linguistic patterns which express roles can be defined as in (Cimiano & Wenderoth, 2007), metaphor and simile (Veale & Hao, 2007) or other kind of relationships such as meronymy, holonymy, etc. (Ruiz-Casado, Alfonseca, & Castells, 2007).

Part-Of Linguistic Patterns

Linguistic patterns have been extensively employed to express part-of relationships. As the same way as in (Hearst M. , 1992), other patterns have been defined in order to discover part-of relationships between concepts. Some of the studies aimed to define meronymy relations within text are (Berland M. C., 1999).

Table 4 presents some of the English patterns that represent a part-of relationship between two concepts. In the first column the pattern is defined, the second one contains an example and the third the two concepts that conform the part-of relation (meronym and holonym).

Pattern	Example	Meronym(NP1, NP2)
NP's NP	Car's engine	Meronym(engine, car)
NP of {the|a|an} NP	Screen of the computer	Meronym(screen, computer)
NP in {the|a|an} NP	Radio in a car	Meronym(radio, car)
NP of NPs	Speed of processors	Meronym(speed, processor)
NP in NPs	Cache in processors	Meronym(cache, processor)
NP have|has|had NP	plant has leaves	Meronym(leaf, plant)
NP come|comes|came with NP	Camera comes with lens cap	Meronym(lens cap, camera)
NP feature|features|featured NP	Camera features zoom	Meronym(zoom, camera)

Table 4- Meronym Patterns

All the patterns described have been manually constructed from observations found in natural language texts. They represent domain-independent regular expressions which can potentially be used in any domain of knowledge.

In general, these approaches are unlikely to find a significant amount of matchings for a pair of related concepts if they use those patterns in a limited corpus. More matchings or data would allow recall to be improved and more robust statistical. For this reason, some researchers as (Ruiz-Casado, Alfonseca, & Castells, 2007), (Pantel & Pennacchiotti, 2006) have tried to extent the basic set of patterns detailed in Table 4 by using pre-related concepts and a domain-related corpus to provide the basis for learning regular expressions. The result is an additional set of patterns which, in many situations, includes domain-dependent concepts within the regular expression (e.g. NP is a city in NP, NP is the capital of NP). These approaches try to solve the problem of the sparseness of data presented in (Berland & Charniak, 1999).

3.5.5 Evaluation of Ontology Learning

These approaches are evaluated generally by comparing the produced concept hierarchies against other handcrafted taxonomies (Hotho & Staab, 2004) but also a brief number of algorithms to perform automatically updates and dynamical conflicts resolution without the control of a human expert have been developed (Ovchinnikova & Kühnberger, 2006).

3.6 Conclusions of the State of the Art

In this section are presented the conclusions after analysing previous contributions in the area of ontologies and specifically in relation to the functions of similarity between concepts and machine learning mechanisms of ontological knowledge.

Related work shows that at most two dimensions of knowledge are used to calculate the similarity measure between concepts, the *is-a* and the part-of relations. However, some ontologies designed previously as (Calle, Castro, & Cuadra, 2008) define more relations between concepts that could provide better results in the calculation of the semantic similarity.

Respect to the evaluation of the similarity measure, it can be concluded that human reasoning is one of the most widely-used methods of comparison when performing validation of a similarity measure. For this reason, such a methodology has also been used in the experimentation section of the present study. Since it is difficult to run a user-based evaluation with complicated ontologies, for example, the Gene Ontology (Lord, Stevens, Brass, & Goble, 2003), it has been deemed necessary here to find or model an ontology with elements that test subjects could understand. Therefore, once the ontological module is implemented, it must be populated with a sufficiently good coverage of domain knowledge, that is, enough knowledge to meet the system requirements.

Regarding to the automatically learning of concepts, there is much related work that describes automatic techniques for learning concepts in order to avoid the manual development and knowledge updating. The most of these approaches are centred in the extraction of different kind of concept relations directly from the text, typically taxonomical relations. Only a few studies report dialog models supported by ontologies, and none of them proposes a system for learning concepts through dialogue.

4 Objectives and Proposal

The objective of this work is to develop an ontological knowledge model for the human-like interaction which will have several strong capabilities. In the first place, Section 4.1 introduces an ontological model based on different dimensions of knowledge. Secondly, in Section 4.2, a similarity function which measures the degree of similarity of concepts using each of the dimensions of knowledge will be presented.

The manual development of large ontologies has been proven to be a very tedious, time-consuming and expensive task. Apart from this, as is well known, every knowledge-based system suffers from the so called *knowledge acquisition bottleneck*, i.e., the difficulty to actually model the knowledge relevant for the domain in question. For this reason, it is necessary to search for an automatic or semi-automatic alternative to populate the ontology. Section 4.3 will present the method designed to this end.

4.1 Ontology Design and Development

The ontological design it is based on several knowledge dimensions that are described in an earlier study (Calle, Castro, & Cuadra, 2008). In this approach, the conceptualization comprises seven ontological dimensions: semiotic, sort (*is-a*), compositional *(part-of)*, essential, restrictive, descriptive, and comparative. The first three dimensions have been previously applied in related works. Essential, restrictive and descriptive dimensions are part of the nature of the concept, they can influence human judgment of similarity and the seventh one, comparative dimension, is derived from previous dimensions and is in charge of calculating the degree of similarity between ontological concepts.

In the following subsections, each dimension will be individually defined and several examples will be presented in order to assist the reader in understanding them. Identifiers to be displayed in each of the examples correspond to the actual identifier of the concept in the WordNet database.

4.1.1 Semiotic Dimension

The *semiotic dimension* (see Figure 10) represents the relationship between concepts, terms (words or collocations) and languages.

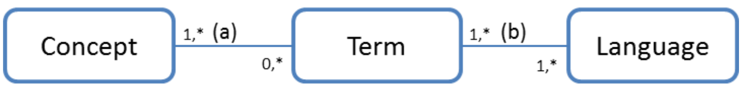

Figure 10 - Semiotic dimension

Firstly, there will be described the (a) relationship between the term and concept. On one hand, a concept can be expressed through a number of terms that goes from 0 to N and on the other hand, a term may refer to a number of different concepts that goes from 1 to M. It should be noted that there may be concepts in the ontology that cannot be expressed by any terms.

The (b) link represents the relationship between terms and languages. The same term can be related to different languages. For example, "radio" is a word in the Spanish language and in the English language.

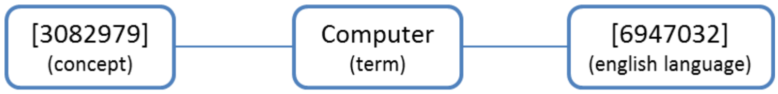

Figure 11 - Example of semiotic dimension representation

For example, as shown in Figure 11, the concept of the WordNet's synset 3082979 corresponds to a machine that is able to perform calculations automatically, and one of the terms associated with this concept is "computer". Other terms related to this concept are "computing machine", "computing device", "data processor", "electronic computer" and "information processing system", all of them also linked to the concept that corresponds to the English language (synset 6947032).

4.1.2 Sort Dimension

The *sort dimension* (see Figure 12) represents the hierarchical structure between concepts. It is also called hyperonymy, hyponymy or *is-a* relation. It links more general concepts like "furniture", "piece of furniture" to increasingly specific ones like "bed", "chair". The sort dimension relates each concept with other concepts with a more specific meaning and models a polytree structure.

All concepts are connected by one or more *is-a* link or a chain of several *is-a* links to the root of the polytree. The *is-a* links point upwards. If an *is-a* link points from a concept X to a concept Y that means that every abstract or real thing that can be called an X also can be called a Y. In other words, every X *is-a* Y. By this way, on one hand a concept may have none, one or more specific concepts (children) and, on the other hand, may be associated with one or more general concepts (parent).

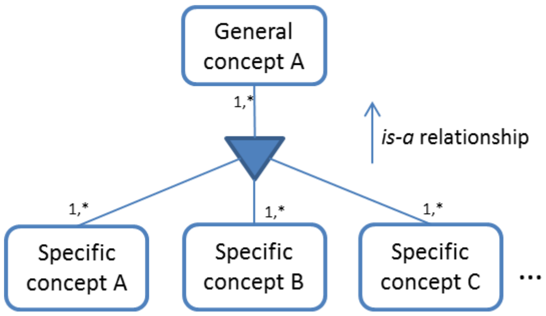

Figure 12 - Sort dimension representation

For instance, as shown in Figure 13, the terms "node", "server" and "web site" are related to concepts that are instances of "computer". The numbers included in the figure correspond to the synset identifiers extracted from WordNet 2.0 database.

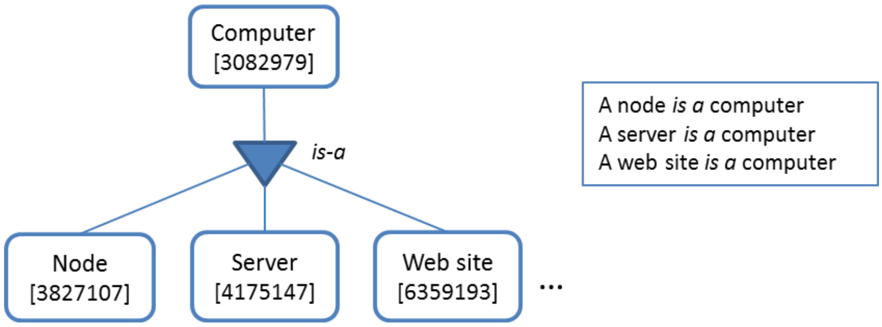

Figure 13 - Sort dimension example

4.1.3 Essential Dimension

The *essential dimension* represents the general taxonomy of concepts. This taxonomy is located in the nodes at the top of the polytree represented in the sort dimension and the root node is the concept associated with the term “concept”. Therefore, the relations included in its design are already observed in the sort dimension. But since they organize the knowledge at the higher abstraction level (they are more discriminative) they should be taken into account separately, adding extra value to similarity measure.

Its design is crucial for attaining good similarity measures, and determines the usefulness of this dimension. The essential dimension of WordNet, for example, classifies the concepts into four main linguistic categories (verb, noun, adjective, and adverb). Such approach is the most adequate for a linguistic interaction domain, but may be weaker in a general interaction domain. This proposal includes an essential design inspired in previous (Calle, Castro, & Cuadra, 2008) and related works (Gee, 1999); (Miller G. , 1995) and refined through preliminary experimentation. The design departs from three main categories (abstract, actions and entities) and develops main classes of concepts, as shown in Figure 14. Finally, it should be added that this proposal is aimed to general interaction domains, and could be improved if suited to specific domains for particular interaction systems.

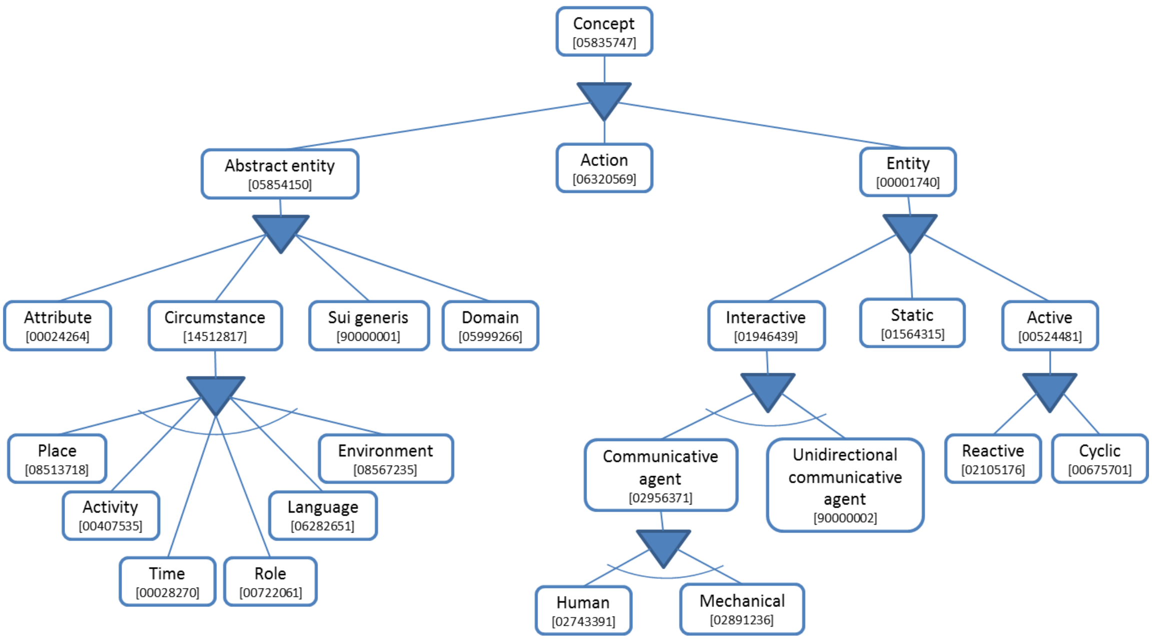

Figure 14 - Essential dimension taxonomy

4.1.4 Compositional Dimension

The *compositional dimension* represents the part-whole relationship between concepts, also called meronymy. A meronym denotes a constituent part of, or a member of something. That is, X is a meronym of Y if Xs are parts of Y(s), or X is a meronym of Y if Xs are members of Y(s).

For example, "leg" is a meronym of "chair" because a leg is part of a chair. Parts are inherited from their superordinates: if a chair has legs, then an armchair has legs as well. Parts are not inherited “upward” as they may be characteristic only of specific kinds of things rather than the class as a whole: chairs and kinds of chairs have legs, but not all kinds of furniture have legs.

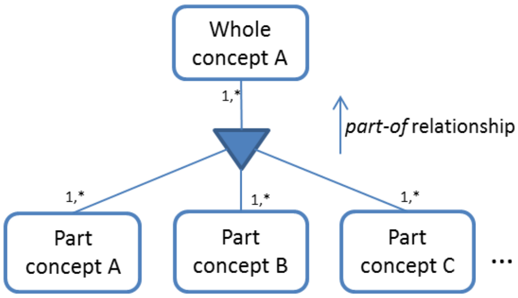

Figure 15 - Compositional Dimension representation

In this way, any concept can have relationships with a collection of concepts that are part of it. Figure 16 shows some of the concepts that are part of a computer, for example "Hard disk", "RAM" and "ALU".

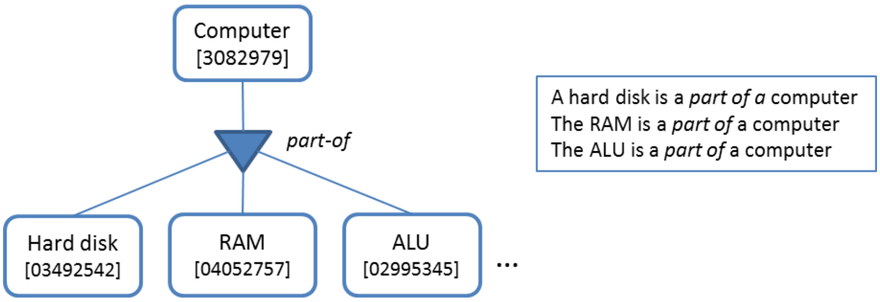

Figure 16 - Compositional dimension example

4.1.5 Restrictive Dimension

Action concepts can be considered a core entity of the semantic interpretation of natural language sentences due to the eminent status of the central verb of a natural language expression. Actions can then be seen as functions that can or cannot be applied to abstract or physical entities, and in the same way, abstract or physical entity concepts can be described through the set of actions that are related to them.

The *restrictive dimension* shown in Figure 17 describes the compatibility between concepts related to some action.

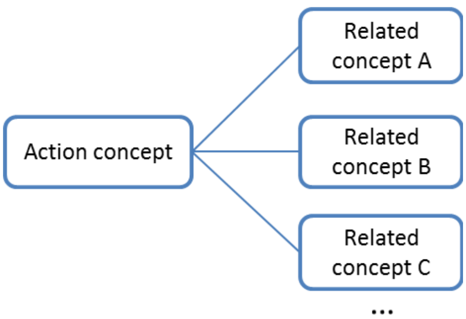

Figure 17: Restrictive dimension representation

For example, Figure 18 shows the action "to compute", which is related to the concepts "computer", "calculator" and "laptop", among others. However, the action "compute" cannot be applied to other concepts, like "cup", "ring" or "beach".

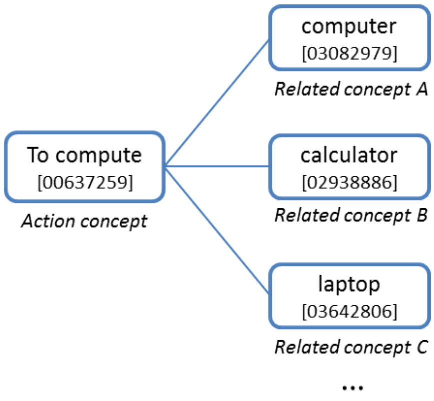

Figure 18: Restrictive dimension example

4.1.6 Descriptive Dimension

The descriptive dimension shown in Figure 19 is in charge of the relationships between three kinds of concepts: generic concepts, attributes and value domains.

The (a) relationship establishes the relation between generic concepts (entities, abstract entities or actions) and attributes that are inherent characteristics of those generic concepts and help further describe or define them. All entities within a given entity type share the same potential attributes.

Value domains are the collections of values which may be assumed by an attribute. The (b) link represents the relationship that attributes have with the set of domain values in which can be represented. Notice that there could be several available domains for a given attribute and that a domain could be numeric (magnitudes regarding a unit) or enumerated (a concept which is *composed* of a set of named values which are also concepts).

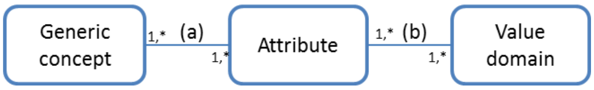

Figure 19: Descriptive dimension representation

An example of this dimension is shown in Figure 20. An instance of the generic concept “hard disk” will have a value in the numeric domain of “Bytes” for the attribute concept “storage capacity”.

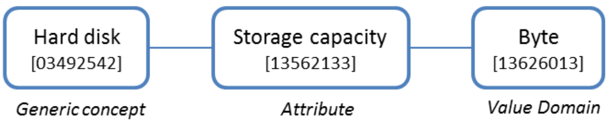

Figure 20: Descriptive dimension example

4.1.7 Comparative Dimension

The *comparative dimension* is derived from previous dimensions and is responsible for calculating in real time the degree of similarity between ontological concepts.

Finally, for reasons of efficiency, most frequently requested similarities can be buffered, that is, stored when calculated, periodically updated and retrieved when necessary.

4.2 Semantic Similarity Measure

This section proposes and evaluates a similarity measure based on the combination of individual similarity measures according to each of the dimensions explained (see Section 4.1). This calculation will take into account the equations applied for each dimension that will be weighted and aggregated in order to obtain a global similarity measure.

The following step is to describe the similarity measure adapted to the described ontological dimensions except for the semiotic dimension. Yet not the only approach, similarity in the semiotic dimension, or similarity between terms is frequently described as the *edit distance* or Levenshtein distance (Levenshtein, 1966), that is, the number of changes necessary to turn one string into another string. The decision to leave this dimension apart is supported by preliminary studies in which this measure yields an average error rate above 50% and in some cases over 80%. Furthermore, for every concept in that study, the accuracy provided by this dimension was lower than that of some of the other dimensions (the semiotic dimension never produced the best prediction), being the only dimension which never ranked first when tested separately. For this reason, it is estimated as it cannot contribute positively to the results (at least, it cannot until it is properly adapted). Last but not least, during preliminary experimentation of the training including this dimension, it was observed that each weight tended to zero, and with the drawback of slowing down convergence of the weights of the rest of dimensions. However, as further work, some evolution of this similarity measure (supported by knowledge on this dimension) can be incorporated into the global measure of similarity.

4.2.1 Inference Mechanisms

This sub-section describes the method used to calculate the degree of similarity between two given concepts in an ontology. Since ontological knowledge here is structured into different dimensions, the similarity measure will also be based on these dimensions. Therefore, partial similarity calculations will be made for the sort, essential, compositional, restrictive and description dimensions described previously. The resulting overall similarity between the two concepts is obtained through the calculation of the weighted average of the five partial similarities

$$S = \frac{S_s w_1 + S_c w_2 + S_e w_3 + S_r w_4 + S_d w_5}{w_1 + w_2 + w_3 + w_4 + w_5}$$

where S_s, S_c, S_e, S_r and S_d are the similarity measures according to the sort, compositional, essential, restrictive and description dimensions, respectively. The values w_1, w_2, w_3, w_4 and w_5 represent the weights assigned to each dimension such that the resulting total similarity between the two concepts will be a value between 0 (completely different concepts) and 1 (the two concepts are the same).

The following sections describe in detail the procedures developed for the calculation of each of the partial similarities.

4.2.1.1 Similarity According to Sort Dimension

The sort dimension represents the *is-a* relationship between concepts. This dimension has a polytree structure, allowing a concept to be a descendant of more than one concept. Similarity in this dimension is often calculated as proportional to the intersection of the list of predecessors of both compared concepts regarding the total size of these lists. To define this measure, a variation of the edge-counting technique – concretely, the conceptual similarity measure defined in the work of (Wu & Palmer, 1994)– has been employed. Given two concepts, C_1 and C_2, this measure can be defined as

$$Sim_s = \frac{2N_3}{N_1 + N_2}$$

where N_1 and N_2 are the number of ancestors of C_1 and C_2, while N_3 is the number of common ancestors of C_1 and C_2 (in the most advantageous tree if several are found in the polytree).

4.2.1.2 Similarity According to Compositional Dimension

The compositional dimension represents the *part-whole* relationship between concepts. For this reason, the most appropriate way to calculate the similarity between two concepts based on this dimension is through the comparison of the parts (or ingredients) of these concepts. Furthermore, the calculation must also take into account the fact that a concept may consist of required and optional concepts. This detail is important when calculating similarity since a greater weight must be given to the required ingredients appearing in both concepts, while a lower weight is given to the optional ingredients. The resulting similarity of two concepts, C_1 and C_2, in terms of the compositional dimension is obtained by applying the formula:

$$Sim_c = \frac{{}^{N_1}/_{M_2} + {}^{N_2}/_{M_1} + {}^{2N_3}/_{(M_1 + M_2)} + {}^{2N_4}/_{(M_3 + M_4)}}{4}$$

where N_1 is the number of common components arising from the intersection of all components of concept C_1 with those components of concept C_2 of type *required*; N_2 is the number of common components arising from the intersection of all the components of C_2, with those *required* components of C_1; N_3 is the number of *required* components that both C_1 and C_2 have in common; and N_4 is the total number of common components (both *required* and *optional*) of the two concepts; M_1 and M_2 represent the number of *required* components of concepts C_1 and C_2, respectively. Finally, M_3 and M_4 indicate the total number of components that C_1 and C_2 have.

4.2.1.3 Similarity According to Essential Dimension

The essential dimension contains a set of abstract concepts which define generic types of concepts (such as action, entity, abstract, circumstance or attribute). This generic classification frequently influences human speakers when estimating similarity. Some other works on similarity calculation posed that concepts are only comparable if included in the same category of WordNet's taxonomy (Wong, 2008). Such approach endows a critical value to this dimension, while omitting the rest of the classification. What is proposed here is that this dimension can contribute to similarity estimation as any other (albeit with a certain weight that could be different than the rest), and that all the concepts observed in the design of the essential dimension may influence the similarity estimation.

The method for calculating similarity between two concepts C_1 and C_2 in the essential dimension is based on the intersection of their essential ancestors (ancestors within the subset of essential concepts). This is formalized as follows:

$$Sim_e = \frac{2Card(E_1 \cap E_2)}{Card(E_1) + Card(E_2)}$$

where $Card(E_1)$ and $Card(E_2)$ are, respectively, the total number of essential ancestors of concepts C_1 and C_2, while $Card(E_1 \cap E_2)$ indicates the number of common essential ancestors.

4.2.1.4 Similarity According to Restrictive Dimension

The restrictive dimension is defined between a concept representing an action and another concept representing an entity. Similarity in this dimension is calculated in a different way depending on the type of concepts to be compared. For this reason, two different similarity measures exist for the dimension: comparing two actions and comparing two entities. Similarity between two concepts representing an entity will be based on the action concepts that both entities have in common. The formula used for the calculation of this similarity when comparing two entities, C_1 and C_2, is defined as

$$Sim_r = \frac{{}^{M_1}/_{(N_1 + N_3)} + {}^{M_2}/_{(N_2 + N_4)}}{2}$$

where M_1 and M_2 are the number of common actions that have a positive or negative restrictive relationship with the entities C_1 and C_2, respectively. The values N_1, N_2, N_3 and N_4 represent, respectively, the total number of actions having a positive relationship with the entity C_1, a negative relationship with C_1, a positive relationship with the entity C_2, and a negative relationship with C_2.

As regards the similarity between two concepts representing an action, this is calculated based on the set of concepts defined on these actions, being more similar the higher the number of restricted concepts in common. The formula to calculate the similarity between two action concepts (C_1, C_2) of a particular sign (positive or negative) is defined as

$$Sim_r = \frac{2N_3}{(N_1 + N_2)}$$

where N_3 is the number of common entities shared by the two actions, and N_1 and N_2 are the total number of entities having a restrictive relationship with C_1 and C_2, respectively.

4.2.1.5 Similarity According to Descriptive Dimension

The description dimension represents the relationship between a concept, an attribute and a value in a concrete domain. Similarity in this dimension is calculated differently depending on the type of concepts to be compared, that is, entities, attributes or domains. For pairs of concepts (C_1, C_2) representing an entity, the applicable formula is defined as

$$Sim_d = \frac{2N_1 + 2N_2 + N_3}{(M_1 + M_2)}$$

where N_1 is the number of common attributes without a default value assigned, N_2 is the number of common attributes whose value is the same for both entities and has not been assigned by default, and N_3 is the number of common attributes with the same value where one of them has been assigned by default. The terms M_1 and M_2 correspond to the total number of attributes related to the concepts C_1 and C_2, respectively.

If both concepts (C_1, C_2) are attributes, the formula to apply is defined as

$$Sim_d = \frac{2N_3}{(N_1 + N_2)}$$

where N_3 is the number of common values of both attributes, and N_1, N_2 is the total number of possible values which can have the attributes C_1 and C_2, respectively.

Finally, if the concepts to be compared (C_1, C_2) represent domains, the similarity according to this dimension is calculated based on the amount of common attributes (for which those domains apply) and the number of values shared by both domains.

$$Sim_d = \frac{{}^{2N_3}\!/_{(N_1 + N_2)} + {}^{2M_3}\!/_{(M_1 + M_2)}}{2}$$

where N_3 is the number of common attributes shared by the domains (C_1, C_2), and N_1, N_2 are the total number of attributes associated with them. Finally, M_3 is the number of common values defined in both domains, and M_1, M_2 are the total number of values of the two domains.

Finally, the concepts to be compared (C_1, C_2) may be values belonging to a domain, either enumerated or of a numeric type. For operating domains, it is necessary to define previously a correspondence between them. Numeric domains can be related through a function (typically, a lineal proportion). Relating an enumerated domain to a numeric domain can be achieved by assigning to each enumerated value a fuzzy label in the numeric domain. Finally, the correspondence between two enumerated domains always involves an intermediate numeric domain (with a correspondence defined to each of the two other domains). Once the values are comparable, the formula to measure their similarity is defined as follows:

$$Sim_d = 1 - \frac{|C_1 - C_2|}{|C_{inf} - C_{sup}|}$$

where C_{inf} and C_{sup} are, respectively, the lower limit and the upper limit within the range of values, and C_1 and C_2 are the correspondent numeric comparable values.

The criteria used to calculate the weights of the ontological dimensions may be diverse. In a first approach, on the one hand this work will be focused on studying the dependence of these weights on the nature of concepts, either in pairs of them (pair training method) or individually (concept training method). On the other hand, it will be explored the influence of the past behaviour of users who perform the concept pair evaluations. Following this line, a user dependent training it is proposed, and finally a hybrid one, merging concept training and user training benefits will be included too. All of them will be evaluated and compared in order to ascertain which one performs better.

4.2.2 Weights Training Methods

Assigning the proper weight to each dimension is crucial to achieving good results. Since a human test subject does not usually give the same relevance to the five dimensions of similarity, a basic training program regarding the weights associated with each dimension was developed. This program is based on the reinforcement learning technique (specifically a variant of the Q learning algorithm) and it has been implemented in order to determine, through several iterations, the appropriate value of the weights applied to each dimension (previously defined in Section 4.1) to minimize the error between the formula result and each human judgment. Therefore, the input to the training algorithm is the set of similarity judgments made by human test subjects. This algorithm follows the next steps:

a) An initial step, where the five weights w_1, w_2, w_3, w_4 and w_5 applied to each dimension (see formula in Section 4.2.1) are initialized to 1.

b) For each iteration of the training algorithm, results for each dimension of similarity are calculated according to the formulas described in the Sections 4.2.1.1 to 4.2.1.5. Subsequently, the five new weights are calculated according to the next criteria:

$$1)\ if\ \forall i\ Sim_i\ < Y\ \rightarrow\ \Delta w'_{\max(Sim_i)} = 1$$

$$2)\ if\ \forall i\ Sim_i\ > Y\ \rightarrow\ \Delta w'_{\min(Sim_i)} = -1$$

$$3)\ Failure\ to\ meet\ conditions\ 1)\ and\ 2) \rightarrow\ \Delta w'_i = \alpha(Y - Sim_i)\Delta w_i Sim_i$$

where parameter i is ranged from 1 to 5 (one for each dimension), represents each individual score and Y represents a similarity value from 0 to 10 for one pair of concepts scored by one of the participant. $\Delta w'$stands for the increase of the weight (for the dimension i) at the current iteration, while $\forall i$ represents that increase at the previous iteration. The $\max(Sim_i)$ and $\min(Sim_i)$ represent the maximum and minimum similarity individual values, respectively.

The training can be focused on different points of view, which will be described in Section 5.1

4.3 Bidirectional Learning of Ontological Knowledge

The second part of this thesis deals with the creation of a system that addresses the learning of new concepts and relations in two directions: from ontology to the user and from the user to the ontology.

In the first case, it is necessary the development of a machine learning mechanism that will allow the automatic population of the ontology through the conversation with humans. The second case presents the opposite functionality, i.e., through dialogue, a mechanism will be able to teach the meaning of a concept to a human. This can be addressed using the knowledge stored in the ontology and some learning services that has been developed using the similarity function which is the first objective of this doctoral thesis.

In practice, for both learning processes (human-to-machine and machine-to-human) a communication interface is required to exchange knowledge. This interface consists of a dialogue system between the user and the ontology, therefore an ontology-dialogue system integration is required.

4.3.1 Automatic Ontology Population through User Knowledge

The automatic population of an Ontology is a major challenge for the engineering knowledge that can be addressed in different ways. This thesis proposes the automatic feeding of an ontology for human-like interaction systems through conversations held with the user. This work aims to demonstrate that, through the processing of sentences in natural language provided by users, the system is capable of creating automatically new terms, concepts, and relations in the ontology.

The type of knowledge acquired by the ontology will depend on the subject of dialogue with the user. The ontology may be populated from very general ('table') to very specific concepts of a domain ('chipset'). The user profile will influence the kind of knowledge that is acquired by the ontology. Thus, ontology could specialize in one or more specific domains.

To attempt the automatic population of ontology knowledge are needed a set of learning services that will allow, on the one hand, to learn new concepts, terms and relations and in the other hand, manage the learned knowledge to ensure the information consistency. In Figure 21 is shown the ontology learning process from the user input to the ontology knowledge base population.

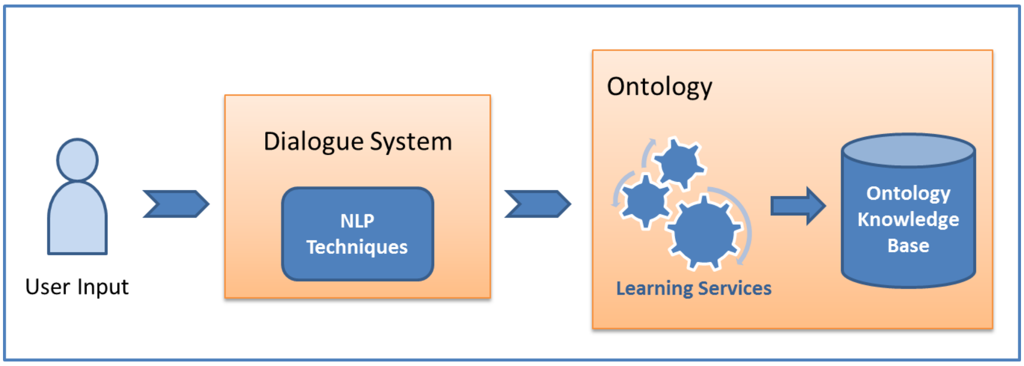

Figure 21 - Ontology Learning Process

In the first place, the user input is received through a chat interface and it is processed by a dialogue system which includes the following NLP techniques by order (see Figure 22):

1) **Break Phrases:** Every time punctuation marks like comma (,), period (.), and colon (:) or semi-colon (;) are found, the sentences introduced by user are divided into subsentences and analyzed separately.

2) **Syntactic Parsing:** Two activities are executed by this module, the lexical tagging and the dependence analysis. In this phase, a specific NLP software is used, in this case (The Apache Software Foundation, 2010).

3) **Semantic Parsing:** The results of the activities carried out by the syntactic parsing are used in this phase, which executes the following tasks: term extraction and relation extraction.

   a. **Term extraction:** In this phase, the terms that are fit to be concepts of the ontology are extracted. On one hand, terms classified in the previous syntactic parsing as prepositions, conjunctions and articles are discarded. On the other hand, terms classified as nouns, verbs, adjectives and adverbs are indicated as possible concepts of the ontology.

b. Relation extraction: The relations between terms are verified and validated through verbs found in the sentences and the natural language patterns observed resulting from the syntactic parsing. This module detects the terms and the relations between them in all the ontological dimensions described in Section 4.1.

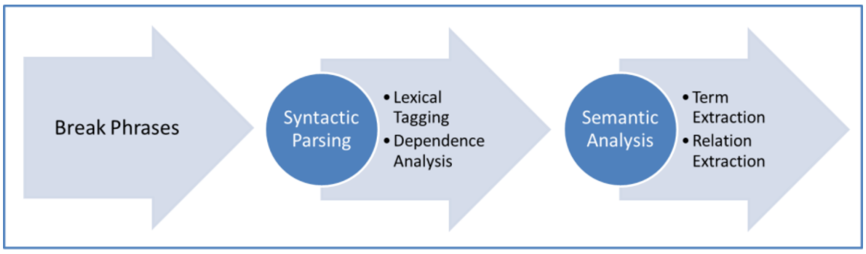

Figure 22 - NLP techniques

4.3.1.1 Ontology Learning Services

The learning services proposed are based in (Calle, Castro, & Cuadra, 2008). In the following sub-sections, for each learning service (from now LS), their purpose and functionality will be explained.

LS1) Terms learning service

When the interlocutor uses a term which is not registered in the ontological knowledge base, the system will ask the user to describe the term. The system will analyze this description and, if the terms expressed by the user contain a synonym already known by the ontology, the new term will be created and associated with the concept related with that synonym.

In the description provided by the user may appear another unfamiliar terms. In this case, the system will be adapted to ask for the meaning of them and learn recursively all the information.

LS2) Concepts learning service

When the interlocutor uses a term which is not registered in the ontological knowledge base and the terms expressed by the user does not contain a synonym already known by the ontology, this learning service will create the new term and a new concept related to that term.

LS3) Relations learning service

The system will be able to add new relationships between existing concepts in the ontology from the information that the user provides during the conversation. These relationships between concepts that can be learned by the system are based on the ontological dimensions discussed previously.

LS4) Unification and dissociation service

The terms, concepts and relationships learned from conversations with users will be monitored regularly in order to detect inconsistent knowledge. To address this issue it could be necessary to perform the unification or dissociation of concepts.

4.3.2 User Learning through Ontology Knowledge

In this section it is described the learning of concepts in the opposite direction: from the ontology to the user. To attempt this objective, a unique learning service will be developed: the concept description generation.

If the user does not know a term used by the system during the conversation, the system will generate a description of the term using the ontological knowledge and the relationships with other concepts based on the ontological dimensions described in Section 4.1, for example, synonyms, hypernyms, meronyms, etc.

This description will be generated with different degrees of detail, depending on user requirements. At first, when the user asks for the meaning of a term, the system will generate a brief description of it based on all the concepts that are related to that term. This description will be generated through the synonyms of that term (if applicable) and by the sort dimension.

Brief description example:

- **User:** What is a camera?
- **System:** The term camera refers to a type of photographic equipment and also refers to a type of television equipment.

If the user requests more information about the term, the system will provide an expanded description using the knowledge stored for the rest of ontological dimensions related to that term: compositional, restrictive and attributive.

Detailed description example:

- **User:** Please, tell me more about the one referred to a type of television equipment.
- **System:** A camera is used to take photographs and consists of a lens system that focuses an image on a photosensitive mosaic that is scanned by an electron beam. It is composed of the following elements: optical lens, a mosaic and a television.

The ontological knowledge gained by the system through the user will support the reversal learning (from system to user), since the system will prioritize concepts expressed by the user before generating descriptions. The generation service descriptions will give more weight to those terms that are more frequent in the expressions that have been learned from the user.

Furthermore, the similarity function defined in Section 1 will be very useful for generating descriptions, since it will be possible to use the concepts in the ontology that are most similar to the concept that is being described.

In this way, one can appreciate the existence of synergies between the similarity function and the learning services. Learning concepts through communication with the user helps to modify the weights of the relations between them and consequently feed the similarity dimension between concepts.

5 Evaluation

In this chapter are presented the methods that will be followed to evaluate, in the first place, the semantic similarity measure proposed in Section 4.2, and in the second place, the bidirectional ontological learning mechanism proposed in Section 4.3.

5.1 Semantic Similarity Measure Evaluation

The next step will be to evaluate the quality of the mechanism developed for the calculation of similarities between two concepts in an ontology which will be specially designed for a human-like interaction system (Calle, 2004).

To evaluate these results, it will be used the comparison with the human judgment. This technique has been applied in several studies about similarity measures and is considered the gold standard.

Once the conceptual model of the ontology has been defined, and the weights training methods proposed, the next step in this study is to evaluate the proposal. The present section describes the experiments run for evaluating the proposal, from their design to the results obtained and discussion.

5.1.1 System Preparation

The knowledge base is supported by the relational database management system Oracle 11g, and the logic of the ontology component (including the inference mechanisms) was implemented in Java. The knowledge bases were designed to satisfy specific purposes within a research project. The initial knowledge load was obtained from the large lexical database WordNet (Fellbaum, 1998) including all the existing concepts (synsets), terms and relationships (corresponding to sort and compositional dimensions). Since the proposed ontological model defines more relationships between concepts (essential, restrictive and descriptive), it is necessary to add more knowledge. The Cognos.Onto tool enables knowledge edition and management for this specific model. This tool belongs to a larger toolkit, Cognos (Calle et al., 2011) already used in several research projects. That toolkit seeks to ease the interaction corpus analysis, annotation, implementation and management, through diverse yet integrated tools aimed to each specific type of knowledge (pragmatic, NLP related, ontological etc.).

5.1.2 Experimental Design and Preparation

First of all, it is necessary to choose an Interaction Domain which will define the entire experiment. The concepts involved will be a subset of the whole knowledge base, restricted to that specific domain. The participants will be chosen in order to constitute a good coverage of the focused domain. Finally, additional knowledge will be fed by experts in that interaction domain not related to the projects where this research is framed (as the test subjects and any other participant in the experiments).

The methodology chosen to evaluate the proposed similarity measure is based on Miller's benchmark (Miller & Charles, 1991). Experiments have been designed to determine whether the result attained through the application of the similarity function on a pair of concepts is reliable or, in other words, if the result falls within an acceptable range when compared with the similarity judgments made by human test subjects.

To begin the experimental phase of the study, an initial loading of concepts must first be made in the proposed ontology. For this reason, WordNet's synsets (Princeton Univ., 2011) were taken as concepts, together with the corresponding semiotics, sort and compositional relationships. Knowledge domain experts have been responsible for populating the remaining dimensions of the ontological model (i.e., the essential, restrictive and descriptive) in a subset of 350 concepts, selected because of their relevance in the interaction domain.

The chosen domain is that labeled as "computer science teaching" interaction domain within the Spanish academic socio-cultural environment. This area of knowledge is familiar to the test subjects who have been selected as heterogeneous in this domain (different roles, ages, and genders). To perform the evaluation, a test was designed for which the test subject had to rate the similarity between pairs of concepts. The set of pairs had to meet a basic criterion: at least two pairs had to be included to explore each of the proposed dimensions, one with clear incidence in the dimension and another one without (or of little impact).

A total number of twenty-one test subjects were available, from which four outliers were left apart. They were discarded after checking their judgment because their responses were not uniform with the rest of the sample. The participant scores follow a normal distribution after removing the outliers. For that reason, the sample size was calculated through a test of statistical significance and the result was at least ten subjects to ensure a 99% confidence. Therefore, a sample size of seventeen participants is sufficient to ensure that the data is representative. The seventeen subjects were all experts in the interaction domain (technical education), specifically five technical students, seven researchers and five lecturers. Their ages ranged from 20 to 50 and were distributed as follows: seven subjects were in the 20-30 year-old range, six in the 30-40 year-old range and the remaining four were in the 40-50 year-old range. With regard to gender, slightly more than half of them were female (9) and the rest were male (8). The chosen interaction domain was the applied on the research project THUBAN (TIN2008-02711). Each participant was provided with a test containing a set of twenty pairs of concepts from this domain. Since the observations follow a normal distribution, it was determined that the minimum significant sample size would be sixteen with 99% confidence. Therefore, a set of twenty pairs of concepts provides significant results. However, in a larger domain, the size of the dataset may be different to attain statistically significant results. In coherence with some other components of the system where this proposal was to be integrated, the similarity measures are ranged from zero (no similarity) to ten (absolutely identical, the same concept). In addition, for each of the pairs, the subjects were asked to justify their score, indicating the specific parameters of similarity that they took into account in making their decision. After obtaining the individual survey results, the average total of the human assessments for each pair of concepts was calculated. Table 5 shows the 20 pairs of concepts included in the test and to the right of each pair, the *range* (difference between maximum and minimum scores), the standard deviation and the average rating assigned by the users.

All the methods are subject to the iteration order (either analyzed pair or human judge), which can alter the result of the training. In order to avoid this effect and to endow significance to the results, through preliminary experiments the minimum number of repetitions (with different order) was determined to reduce stochastic and gain significance (close to 275), and consequently it was decided to program 300 repetitions with a different order for each method. In the graphs and tables, error rates of pairs (identified by *pair_id*) are numbered from 0 to 19, while iterations are numbered from 1 to 20.

Pair ID	Pair of concepts	Range	Standard deviation	Average similarity
0	Reading lamp – Personal computer	6	1.76	2.71
1	Laptop – Server computer	6	1.62	6.47
2	Teacher – Tutorial	7	1.92	5.06
3	Meeting room – Laboratory	8	2.15	4.35
4	Server computer– Microwave	8	2.02	2.24
5	Office – Laboratory	9	2.25	5.76
6	Screen – Blackboard	7	1.83	6.12
7	Stapler – Folder	7	2.19	3.94
8	Plug– Power strip	4	1.21	8.29
9	Office – Meeting room	6	1.69	6.29
10	Pencil – CD marker	3	0.99	7.29
11	Associate professor – Teaching Assistant	5	1.34	8.06
12	Associate professor – Bachelor	8	2.53	5.18
13	To write papers – To program	7	2.15	4.53
14	To give a lecture – To teach	6	1.60	7.76
15	Keyboard – Mouse	5	1.41	7.35
16	Fridge – Microwave	7	1.77	5.35
17	Hard disk drive – Pendrive	3	0.94	8.47
18	Scanner – Printer	8	1.89	5.94
19	Poster – Blackboard	6	1.82	4.24

Table 5- Pairs of concepts and average similarity

5.1.3 Evaluation Methods Design

Four different evaluation methods have been designed to train the weights of the similarity function: pair-oriented, user-oriented, feature-oriented and an hybrid training method.

5.1.3.1 Pair-oriented training method

Firstly, a pair-oriented training was implemented in order to individually adjust the weights for each of the 20 concept pairs, independently of the specific user. The weights are adjusted individually for each of the pairs of concepts, taking one user per iteration. In this way, after each iteration, a new array of refined weights is obtained and used for evaluating the similarity. The test consists of calculating the similarity (with that array of weights) and comparing it with the human assessment.

5.1.3.2 User-oriented training method

Since the degree of significance assigned to each dimension may depend on the subjectivity of the testers, it was of particular interest to make an adjustment of the weights based on each user.

In this experiment, the training of the weights was performed once for each user and consisted of 20 iterations (one for each pair of concepts). For an iteration of this training algorithm, absolute error committed in relation to the corresponding pair was calculated. After running the training for the 17 users, the average of the absolute errors for each of the iterations was calculated.

5.1.3.3 Concept-oriented training method

The third method has been designed in order to address the shortcomings of the pair-oriented training. It should be indicated that storing an array of the weights for each possible pair of concepts in a medium sized ontology requires unusually extensive physical resources. Besides, a significant coverage of the thus defined knowledge would require far too much training. In short, it is not realistic to develop that method because of the high number of combinations of concepts. However, through preliminary experimentation it was checked that the weights applied to a pair were also likely to be applied to other combinations of each of those two concepts. Therefore, a new training method (*feature-oriented*) was proposed by slightly modifying the pair-oriented one. In the *feature-oriented* method, the array of weights is stored for each concept instead of for each pair of concepts (which solve both the problems of storage and the extent of training). Each time one concept is compared to any other, its array of weights will be reviewed and refined. The similarity calculation for a given pair is based on the aggregation of the arrays of both concepts.

5.1.3.4 Hybrid method

Finally, it was observed that each method showed a different behavior depending on the pair of concepts compared: the method achieving the worst results on average was also the best for some specific pairs. Subsequently, a hybrid method was proposed and has been developed, combining the feature-oriented and user-oriented trainings, aiming to profit the advantages of each method. The training will be similar to that focused on the user, but for each iteration the array of weights will be refined to a different degree, taking into account the array stored for each particular concept. Therefore, if a particular dimension is usually relevant for a concept, adaptation to the user in that dimension will be strengthened.

5.1.4 Preliminary Experimentation

Before testing the proposal, some preliminary experiments were performed to refine it and to obtain a first perspective on its validity. These experiments have been instructed on a set of similarity measures obtained from a total of 20 pairs of concepts, described in section 5.1.2 and evaluated by 17 human subjects.

Specifically, the individual influence of each dimension in similarity was tested thorough a set of experiments involving each of them separately. Since there is no combination of them, there is no need for training either. Figure 23 shows a box plot that represents the error measures produced individually for each dimension.

Figure 24 shows that in series of twenty pairs, every dimension produced better prediction than the others at least once. In fact, the essential dimension provided the best response in almost half of the cases, while the descriptive dimension was best in just one case.

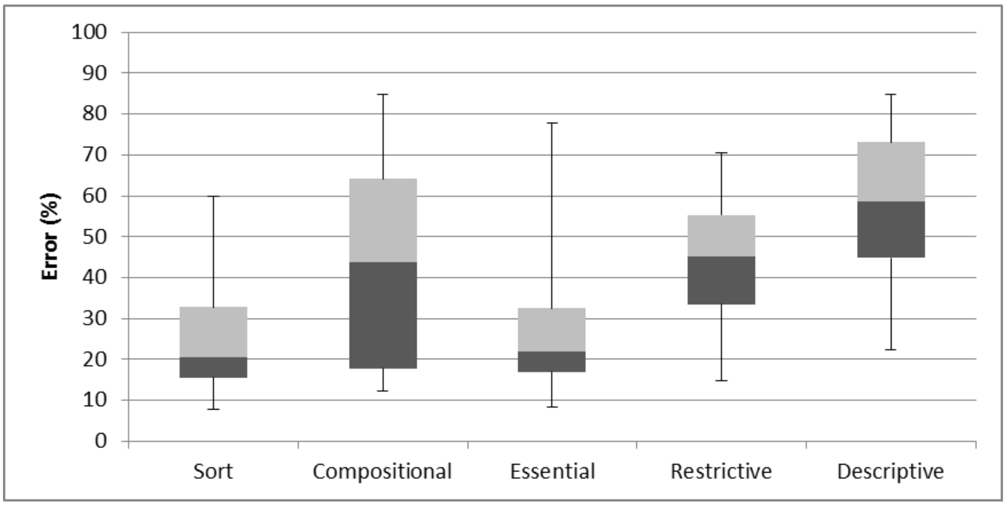

Figure 23 - Performance of each training method

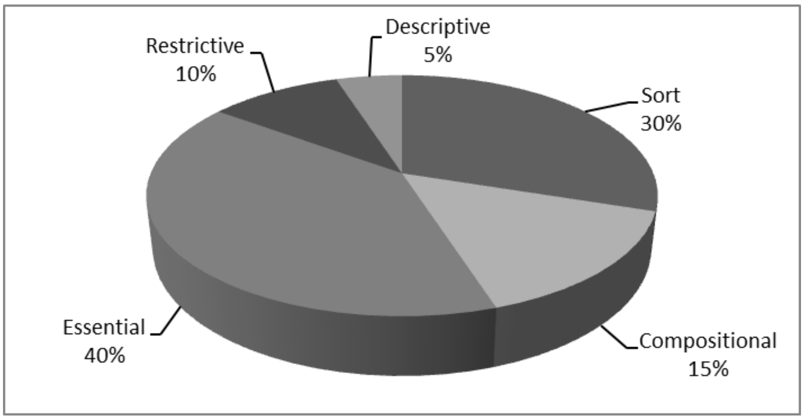

Figure 24 - Cases in which each dimension is ranked first

5.1.5 The experiments

This section presents the results obtained after the execution of the experiments corresponding to the four weight adjustment algorithms described in Section 4.3. These experiments were performed on a subset of the ontological knowledge stored acquired from the computer science teaching domain. The first experiment performed was the pair-oriented training and, in order to evaluate the results of this training, the average of the absolute error was calculated (for each pair) between the similarity based on each human judgment and the result obtained by applying the similarity measure proposed according to the following formula:

$$\overline{error_{pairId}} = \frac{\sum_{i=1}^{n} error_{pairId}(it_i)}{n}$$

where *i* corresponds to an index to iterate over each human judge for a specific pair of concepts and *n* is the number of test subjects. Finally, $error_{pairId}$ represents the absolute error between the human judgment for that pair and the result obtained through the training algorithm in that iteration. Table 6 shows the absolute errors calculated in this experiment for each pair of concepts, as well as the average error which, at about 18.5% comes slightly closer to the scores provided by the human subjects.

It should be noted that in eleven cases, the error rate is less than the average, in eight cases the error rate is around the average, and one pair (#2) shows an excessive error rate that requires further analysis and discussion (see subsection 5.1.6). Figure 25 shows a comparison of the trend lines regarding the error rate accumulated by the pair-oriented training algorithm and the accumulated error by the similarity function without weights training.

	Pair Id																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	AVG
error (%)	15.2	14.8	38.3	18.6	19.4	18.1	17.6	18.8	20.2	15.4	13.4	18.0	22.5	19.6	15.2	13.0	15.3	20.9	17.1	19.0	18.5

Table 6 - Pair-oriented training error rate

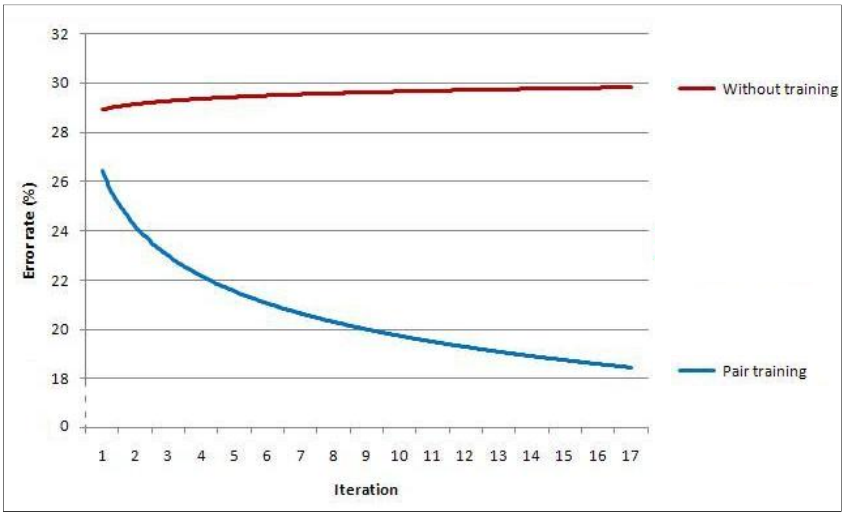

Figure 25 - Accumulated average error in pair-oriented training

In second place, the absolute error obtained for each pair in the feature-oriented training is shown in Table 7. These results, compared with those obtained for the pair-oriented training, show slightly worse performance (with a mean error rate of 20,2%). However, it should be recalled that this method has other advantages (realistic storage and training extent).

	Pair Id																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	AVG
error (%)	15.0	14.9	38.2	18.4	30.3	22.7	17.5	18.5	20.1	21.6	13.4	24.9	21.2	19.2	15.2	13.0	14.4	20.6	16.8	25.4	20.2

Table 7 - Feature-oriented training error rate

The third experiment executed was the user-oriented training. In order to evaluate the results of this experiment, the average of the absolute error was calculated (for each human judge) between the similarity based on each human judgment for the 20 pairs of concepts and the result obtained applying the similarity measure proposed. In this way, the error average has been calculated as follows:

$$\overline{error_{userId}} = \frac{\sum_{i=1}^{n} error_{userId}(it_i)}{n}$$

where *i* corresponds to an index to iterate over each pair of concepts for a specific user, *n* is the number of pairs of concepts and $error_{pairId}$ represents the absolute error between the human judgment for that pair and the result of the training algorithm in that iteration.

In this case, the average error rate achieved is 23.9%, even worse than that for the feature-oriented training. The absolute error rate obtained for each iteration is shown in Table 8.

	Pair Id																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	AVG
error (%)	18.6	14.1	40.7	17.9	30.8	16.8	22.9	17.5	37.1	17.1	34.6	35.8	24.3	21.5	31.2	13.6	13.9	27.4	22.3	20.8	23.9

Table 8 - User-oriented training error rate

Figure 26 shows a comparison of the trend lines correspondent to the error rate accumulated by the user-oriented training algorithm and the accumulated error without any weight training. As can be observed, the user-oriented training trend line follows a downward curve and after 20 iterations reaches an error rate of 23.9%. Comparing both trend lines, it can be concluded that this training decreases the accumulated error and adapts the calculated similarities to the subject's judgments, yet it would be desirable to improve that adaptation (since it is still far from feature-oriented training).

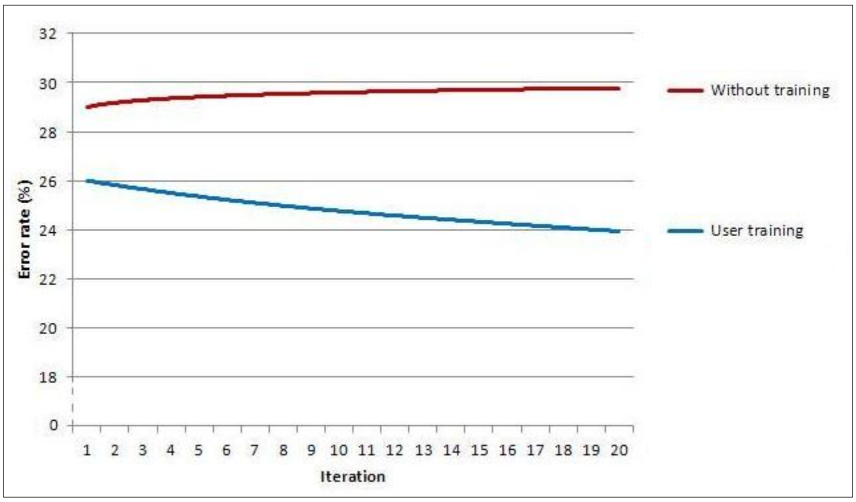

Figure 26 - Accumulated average error in user-oriented training

As observed, both the user-oriented and the feature-oriented training methods are able to improve the similarities calculation, becoming noteworthy approaches. Consequently, it has been found of interest to explore a method which combines both of them. This new hybrid method departs from the user-oriented approach, and takes into account the weights vector obtained from the feature-oriented training described in Section 4.2.2. As shown in Table 9, the user error rate has been successfully reduced to 21.2% with respect to the user-oriented training. However, this method degrades the performance achieved by the feature alone method.

	Pair Id																				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	AVG
error (%)	16.0	14.3	39.0	16.9	29.4	17.3	22.2	17.4	25.1	18.5	21.5	29.6	22.0	19.9	22.8	13.2	13.1	23.8	19.2	22.8	21.2

Table 9 - User-feature hybrid training error rate

5.1.6 Discussion of Results Obtained

From the preliminary experimentation, we can lead to the conclusion that the essential design was appropriate, and that the descriptive dimension was weak. In further analysis it was found that the latter lacked sufficient knowledge, and it was improved in this line before evaluation (more knowledge was added). Despite this improvement, since the analysis and introduction of this knowledge is performed manually (in contrast to other dimensions, for which knowledge was obtained from WordNet), it could still be enhanced and this would improve the individual results of this dimension. Besides, this result is not definitive, since the weights may be different in other interaction domains, and the volume of the knowledge base is important too. But a useful consequence is that each one of the five ontological dimensions can contribute to the similarity function, supporting the hypothesis that an adequate combination of them may yield better results than any of these individual approaches.

Among the results, concept pair 2 (*teacher-tutorial*) scored an error rate above 38% and the average similarity assigned by users (see Table 5) was 5.06. This latter value is significantly high considering the fact that the first concept refers to a person and the second is a static entity. Reviewing participant responses to this question, however, it can be understood that test subjects gave a higher score to the sole feature the concepts have in common, the activity of teaching. Analyzing the results of this outlier, it appears that the algorithm has a tendency to gradually increase the weight of the restrictive dimension, but longer training will be necessary to adapt the weight vector so that the only relevant dimension is the restrictive one. Using a training algorithm with faster convergence would ensure a good result in this pair, but could adversely affect the other results. However, convergence is guaranteed with a larger number of users.

Figure 27 shows the comparison of the absolute error obtained in the four experiments performed in this work (pair-oriented, user-oriented, concept-oriented and hybrid trainings) for each pair, and also the average results of each method. The first experiment performed, the pair-oriented training, achieves the best average error rate, about 18.5%, although in the pair mentioned above the error exceeded 38%. However, this experiment has a major limitation: a trained weight vector for each pair of concepts possible cannot be stored due to the large number of combinations of existing concepts in the ontology. This shortcoming was mitigated with the development of the concept-oriented training, achieving an error rate about 20.2%, a figure which is slightly worse than that of the pair-oriented training error. Nevertheless, this result does not fully reflect the impact of this training because not all test pairs include concepts that appear more than once in the experiment. If the calculation of the average error is restricted to those pairs which have concepts repeated in more than one pair, then the error amounts to 22.8%. In any case, this experiment has an important advantage since its implementation is more realistic and can be applied to large ontologies.

The user-oriented training was aimed at adapting the weights to each subject in order to confirm the assumption that not every test subject assigns the same value to all dimensions. Although the error rate achieved (23.9%) was not as satisfactory as either the pair or the feature-oriented trainings, the figure included in the sub-section 5.1.5 for the training shows a decreasing trend line which, when compared with the trend line without training, allows for the conclusion that the user-oriented experiment is able to adapt to each individual judgment. For this reason, an improvement was attempted with the user-training result through its combination with the feature-oriented experiment.

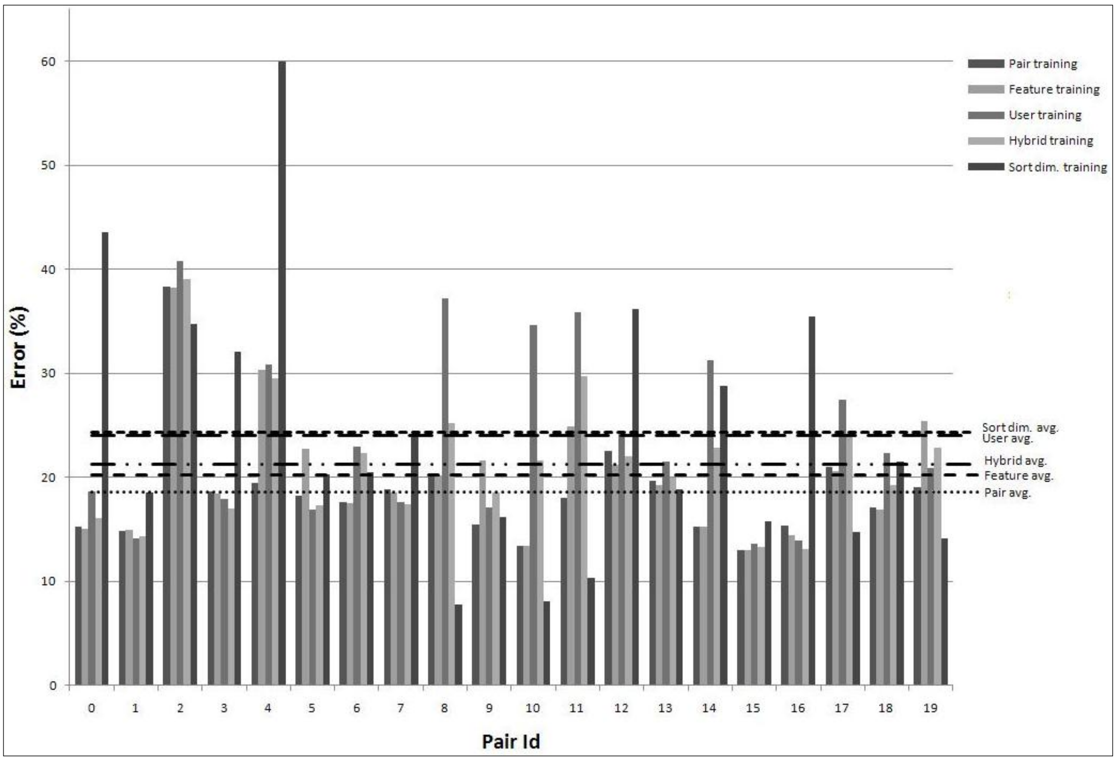

Figure 27 - Comparison of the experiment results

The hybrid training detailed in Section 5.1.3.4 achieved 21.2% in the error rate, which reduces that of the user-oriented training, and balances the performance of the user-oriented method (reduces standard deviation). Taking into account that feature-oriented training method depends on the experience and that for some features the knowledge base might lack of this experience, the response obtained could not be satisfactory in some cases. In fact, when calculating the error produced by the feature-oriented method over the dataset (not restricted to repeated pairs) the result amounted to 22.8%. In sum, the feature-oriented method provides better results but only if enough knowledge is available. The last results presented in Figure 27 concern an experiment observing only the sort dimension (which is a frequent method for calculating similarities). Its average error rate is 24.1%, which is higher than that for any of the four methods discussed. In addition, it can be observed that the error rate of this experiment is, in several cases, far from the average error. Figure 28 shows a boxplot comparing the performance of the four training methods proposed and the sort dimension formula.

As can be seen, regarding the error in the predictions, the sort dimension obtains higher maximum (although also lower minimum), higher median (except for user training) and higher deviation than the rest. From this graph, it can be concluded that the error rate achieved by the sort dimension method (used in previous studies of similarity) is greater than the error rate achieved by the feature training method. In order to check that statistically, a null hypothesis was formulated (the average error is the same in both methods) and also an alternative hypothesis, (the average error of the feature-oriented method is lower than the sort dimension method error).

The measure of discrepancy has been calculated for the sample of twenty measures of error (one per pair) and the result (-1.78) was found to be outside the acceptance range (-1.64, +∞), therefore the null hypothesis is rejected and the alternative is accepted with a significance level of 0.05. Consequently, it is considered true that the error shown by the feature-oriented method is lower than the error produced by the sort dimension method.

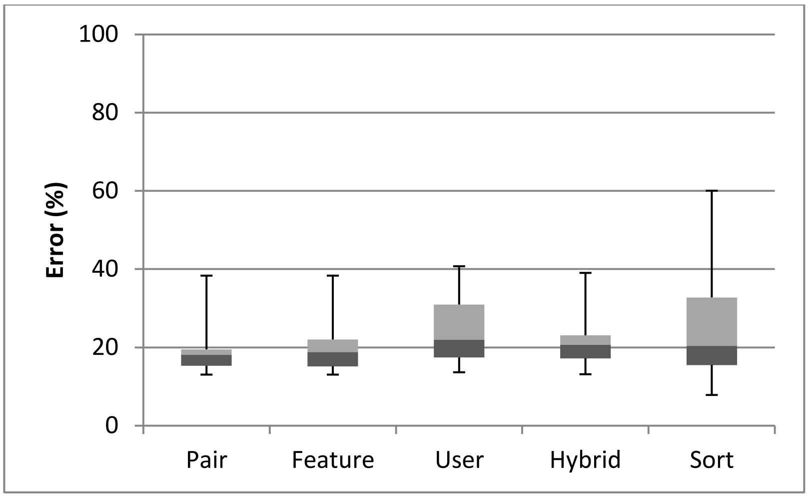

Figure 28 - Performance of each training method

Finally, Figure 29 shows the average final weights of the four experiments. It shows the relevance taken through the experiments by each dimension, yet it cannot be extrapolated to other interaction domains. While dependent on the set of pairs chosen for the experiment, these results show that how all five dimensions are taken into account, with diverse weights.

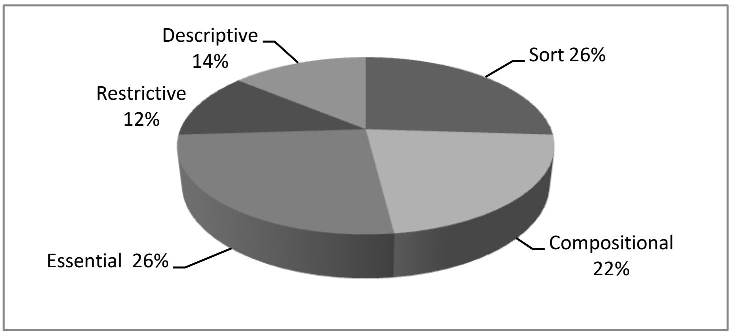

Figure 29 - Average weights of the ontological dimensions

5.2 Ontological Knowledge Acquisition Evaluation

In order to determine the quality of the knowledge acquired by the ontology on one side and by the user on the other side, two different evaluation techniques are needed.

In the first case exposed in Section 4.3.1, the ontology is the learning receiver and the user has the role of trainer. Through dialogue with the user, learning services will enlarge the ontological knowledge base by adding new terms, concepts and relationships.

The automatic acquisition of knowledge by the ontology through the dialogue with humans is a new contribution in the ontological learning area and, as stated in the prior art, there are not many mechanisms for evaluating this kind of systems. The method (Hoto & Staab, 2004) is a good base, but involves high costs and does not provide sufficiently significant results unless it is applied several times. For this reason, it looks appropriate to propose a new one, inspired by the aforementioned but with variations aimed at strengthening these points. In addition, this new evaluation methodology will include the creation of a gold standard to evaluate the results of the automatic ontological knowledge acquisition.

5.2.1 Evaluation Methodology

In this section will be described the technique used to evaluate the automatic acquisition of knowledge performed by the ontology. As has been stated before, it is necessary to design a new evaluation methodology for this purpose, since there are no previous studies on the state of the art to address this purpose.

In many of the techniques used for ontology evaluation, for example in the calculation of similarities between concepts, human judgment is used to obtain a gold standard to measure the quality of the proposed techniques. Moreover, the problem to be solved is human nature itself, so that the proposed methodology will be designed to compare the knowledge acquired by the ontology and knowledge acquired by a human when processing a set of texts.

The group of texts will be chosen randomly from a set of texts in several languages English and Spanish and will belong to different topics and knowledge areas. These texts will be provided to a group of experts with different levels of experience in linguistic annotation and, for each of the texts, experts will write down all semantic relationships belonging to each dimension of ontological knowledge.

Once all the experts have annotated the relationships found in all texts, a meeting of experts will take place. In this meeting the experts will analyze the text jointly, comparing and discussing all the semantics annotated (by themselves and by the system). Once finished, they will have reached a super-set of any annotation, which is commonly accepted by all those experts as the better approach possible. That super-annotation will be taken as the gold standard to perform the evaluation.

The analysis of the annotations will by supported by four main **metrics:**

- **Number of hits:** number of relations found in the text by the expert.
- **Number of absences:** number of relations that has not been found in the text.
- **Number of buggy knowledge:** Number of relations that have been annotated wrong and cannot belong the gold standard.
- **Number of redundancies:** Number of relations with knowledge that can be acquired by other semantic relations or is duplicated.

Once formed the gold standard, its results are compared with the annotations of each expert. Finally, the gold standard will be compared with the results obtained by the automatic acquisition system.

Therefore, the methodology can be summarized in the following phases:

1) Preparation: text selection; experts selection.
2) Training: experts are trained in the task (specific type of annotation).
3) Expert's performance.
4) System performance.
5) Expert's consensus.

6) Comparison of each individual analysis (of each expert and of the system) to the gold standard.
7) Result analysis.

5.2.2 Experimental Design

In this subsection will be described the experimental design, that includes on one hand, the description of the set of texts chosen to create the gold standard and to perform the evaluation and, on the other hand the number and profile of the experts that have been involved in the evaluation.

5.2.2.1 Texts and Users description

The set of texts used to obtain the gold standard and to perform the evaluation were chosen randomly from a set of one hundred books in English and Spanish languages which belong to different topics and knowledge areas, such as science, history or economy. The number of selected texts was nine, five written in English and four in Spanish and the fragment text size rounds between 50 and 80 words.

In Table 10 are described the main characteristics of the texts included in the evaluation.

Text Id	Book Title	Author	Language	Area	Number of words
1	The power of Green	Ironbuttz	English	Cuisine	71
2	El drama humano	Vicente Américo Caballé	Spanish	Science	52
3	Forex Trading Tutorial for Begginers	Sona Matasyan	English	Economy	56
4	Ayúdame a recordarte	Génesys L. Pantoja	Spanish	Fiction novel	62
5	Princess Rose	Holly Jackson	English	Fiction novel	59
6	Empresas de inserción y nuevas oportunidades de negocio	Observatorio de Economía Solidaria	Spanish	Economy	55
7	The Encounter	Idaean Halley	English	Fiction novel	56
8	Introduction to Software Testing	Pawan Kumar Nali	English	Science	52
9	Cocina moderna	Anónimo	Spanish	Cuisine	76

Table 10 - Texts included in the evaluation

The experts that have been involved in the gold standard definition and in the evaluation process have three different levels of experience in linguistic annotation: expert (more than 10 years of experience), medium (between 1 and 2 years of experience) and novel (less than 6 months of experience). In the evaluation have been involved a total of six users, two with a novel level of experience, two with a medium level of experience and two users with high level of experience in linguistic annotation.

5.2.2.2 Instructions to perform the experiment

In the following paragraphs will be detailed the instructions that each user received to perform the experiment.

In the first place, a document with the nine texts belonging to the evaluation experiment was sent to the users. Below each text, users can found a table to write down all the semantic relations between concepts identified for each dimension. Table 11 shows the information provided to the user before beginning the evaluation. It includes the description of the information that should be annotated: the type of relation, the relation existing between concepts and an example for each one.

Semantic Relation/ Dimension	Function	Example
Synonymy	synonymous_of	is_synonymous_of(clever,smart)
Sort	is_a(hypernym, hyponym)	is_a(chair,piece of furniture)
Compositional	part_of(meronym,holonym)	part_of(chair leg, chair)
Restrictive	restricts(action,concept,sign) sign:positive (0) if the action can be perform by the related concept or negative(1) if the opposite occurs	restricts(to teach,teacher,0)
Descriptive	describes(concept,attribute,domain_value)	describes(hard disk drive, storage 1TB)

Table 11 - Semantic relations definition and examples

The users were told to annotate only the relations they can found in the texts, without using their previous knowledge. In this way humans have no advantage over the system and vice versa.

5.2.2.3 Gold Standard definition

Once all individual analysis and the system analysis have been completed, they will be considered by the joint of all experts (both experts and system analysis) following a consensus in order to obtain the gold standard.

		Number of relations for each dimension				
		Semiotic	**Sort**	**Compositional**	**Restrictive**	**Descriptive**
Text Id	**1**	25	10	7	12	2
	2	17	8	5	2	1
	3	13	8	0	3	0

	4	23	0	7	2	10
	5	18	2	0	7	7
	6	23	1	0	7	8
	7	21	6	1	3	5
	8	18	2	1	9	1
	9	28	16	11	4	2
Average		20,67	5,89	3,56	5,44	4,00

Table 12 shows, for each text of the experiment, the number of relations existing for each semantic dimension. These results represent the gold standard that will be used to evaluate the results of each user who has participated in the evaluation and those obtained by the automatic acquisition system.

		Number of relations for each dimension				
		Semiotic	**Sort**	**Compositional**	**Restrictive**	**Descriptive**
Text Id	**1**	25	10	7	12	2
	2	17	8	5	2	1
	3	13	8	0	3	0
	4	23	0	7	2	10
	5	18	2	0	7	7
	6	23	1	0	7	8
	7	21	6	1	3	5
	8	18	2	1	9	1
	9	28	16	11	4	2
Average		20,67	5,89	3,56	5,44	4,00

Table 12 - Gold Standard Values

5.2.3 Experimental Definition

Once all experts have completed the questionnaire writing down the semantic relations found in all the texts, the results will be compared with the gold standard. This task is performed counting the number of relations for each dimension found for each expert in each text and comparing the results with the gold standard defined in Section 5.2.2.3. With this result will be calculated the hit rate of the experts. In addition, are taken into account other two variables: the number of buggy and redundant relationships annotated by the experts.

On the other hand, to evaluate the effectiveness of the ontological knowledge acquisition system, all the texts are introduced into the system which only contains knowledge belonging to the semiotic dimension. Once the texts are introduced into the system, the semantic relationships that can be identified are automatically learnt. After this process, the number of relations learnt by the system for each dimension and for each text are counted and compared with the values defined in the gold standard. With this result will be calculated the hit rate of the system. In addition, the number of buggy and redundant relationships acquired by the system is taken into account in the experiment.

The experimental results will be presented taking into account different factors:

- User experience levels.
- Text languages.
- Text areas.

For each factor, will be calculated the following measures:

- The hit rate, defined as follows:

$$Hit\ rate = \frac{Rf}{Rgs}\ x\ 100$$

where Rf represent the number of semantic relations found and Rgs the number of semantic relations defined in the gold standard.

- The buggy knowledge rate, defined as follows:

$$Buggy\ knowledge\ rate = \frac{BRf}{Rgs}\ x\ 100$$

where BRf represent the number of buggy semantic relations found and Rgs the number of semantic relations defined in the gold standard.

- The redundant knowledge rate:

$$Redundant\ knowledge\ rate = \frac{RRf}{Rgs}\ x\ 100$$

where RRf represent the number of redundant semantic relations found and Rgs the number of semantic relations defined in the gold standard.

5.2.4 Experimental Results

In this section the results of experimentation will be detailed. In the first place, Table 13 shows the average number of relations found in the texts for each semantic dimension according the user experience in pragmatic annotation. Each row shows for each user profile, the average number of relationships for each dimension, and finally the user's average of relations found.

		Average number of relations				
		Semiotic	Sort	Compositional	Restrictive	Descriptive
User	Less than 6	13,83	2,66	2,05	1,94	1,77

experience	**months**					
	Between 1 and 2 years	14,05	3,44	2,72	3,38	1,27
	More than 10 years	19,22	5,22	3,11	4,44	2,66
	User's Average	15,00	3,49	2,53	3,02	1,76

Table 13 – Average number of relations annotated by the users

In the second place, Table 14 shows the average number of relations learnt for each dimension by the system.

	Average number of relations found for each dimension				
	Semiotic	**Sort**	**Compositional**	**Restrictive**	**Descriptive**
System	16,11	3,11	2,67	5,00	2,56

Table 14 - Average number of relations learnt by the system

Taking into account these results, the three measures defined in Section 5.2.3: hit rate, buggy knowledge rate and redundant knowledge rate will be calculated for the system and the experts and then the results will be compared in order to three different factors: user experience levels, text language and text areas.

5.2.4.1 Global Results

In this section are shown and compared the overall results for the hit rate, the buggy knowledge rate and the redundant knowledge rate obtained for the system and for users.

Table 15 shows the hit rate results for the average of users and system. For the three first dimensions (semiotic, sort and compositional), both users and system results are satisfactory (hit rate over 69%). However, for restrictive and descriptive dimensions, in the case of users, the results are not so acceptable.

	Hit rate				
	Semiotic	**Sort**	**Compositional**	**Restrictive**	**Descriptive**
Users	72,58%	69,62%	71,25%	55,51%	43,89%
System	77,96%	72,38%	75,00%	91,84%	63,89%

Table 15 - Hit rate results

In Figure 30 is shown that the hit rate obtained by the system is higher than that obtained by the user's average for the five dimensions. For the first three dimensions the system hit rate is slightly higher and significantly higher for the two remaining dimensions.

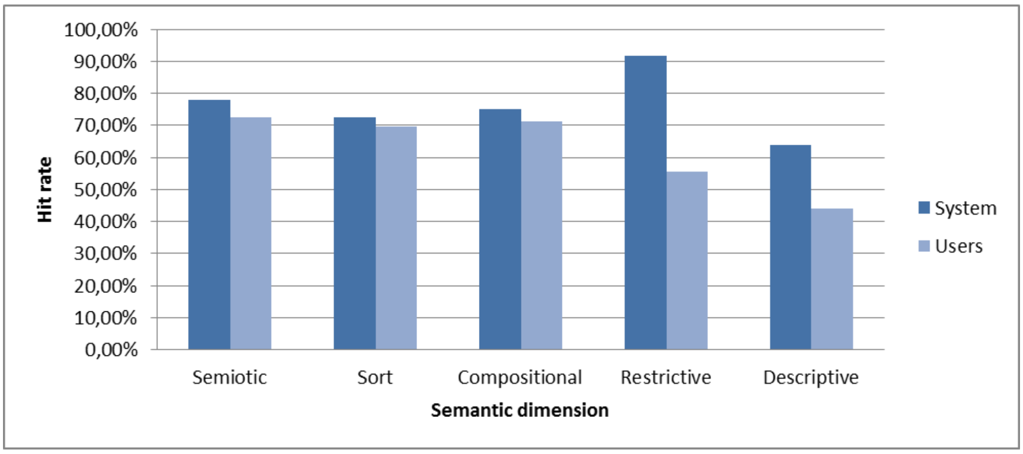

Figure 30- Hit rate per semantic dimension

Table 16 shows the buggy knowledge rate results obtained for the users average and the system. On one hand, users obtain less than a 7% of buggy knowledge in the five dimensions and on the other hand, the system obtain a buggy knowledge rate less than 10% in all the semantic dimensions except for the descriptive dimension.

	Buggy knowledge rate				
	Semiotic	Sort	Compositional	Restrictive	Descriptive
Users	1,83%	0,25%	5,63%	6,53%	4,44%
System	2,15%	0,00%	9,38%	8,16%	27,78%

Table 16 - Buggy knowledge rate results

In summary, the results of buggy knowledge shown in Figure 31 are a little higher for system in the first four dimensions and significantly higher for the descriptive dimension.

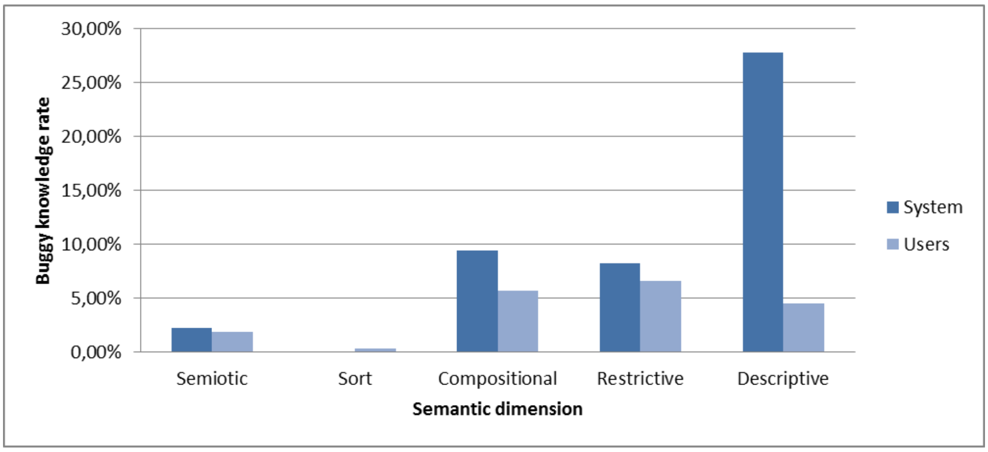

Figure 31- Buggy knowledge per semantic dimension

The results shown in Table 17 are related to the redundant knowledge rate obtained by the user's average and the system. The redundancy rate is low for semiotic, sort and restrictive dimensions for both, users and system. However, compositional and descriptive dimension present higher redundant knowledge rates, between 12% and 23%.

	Redundant knowledge rate				
	Semiotic	Sort	Compositional	Restrictive	Descriptive
Users	0,54%	0,84%	13,75%	0,82%	14,44%
System	1,08%	0,00%	12,50%	6,12%	22,22%

Table 17 - Redundant knowledge rate results

Figure 32 shows the redundant knowledge rate comparison between system and user. It can be seen that for semiotic and restrictive and descriptive dimensions the system rate is higher than user's rate, while for the sort and compositional dimensions the opposite occurs.

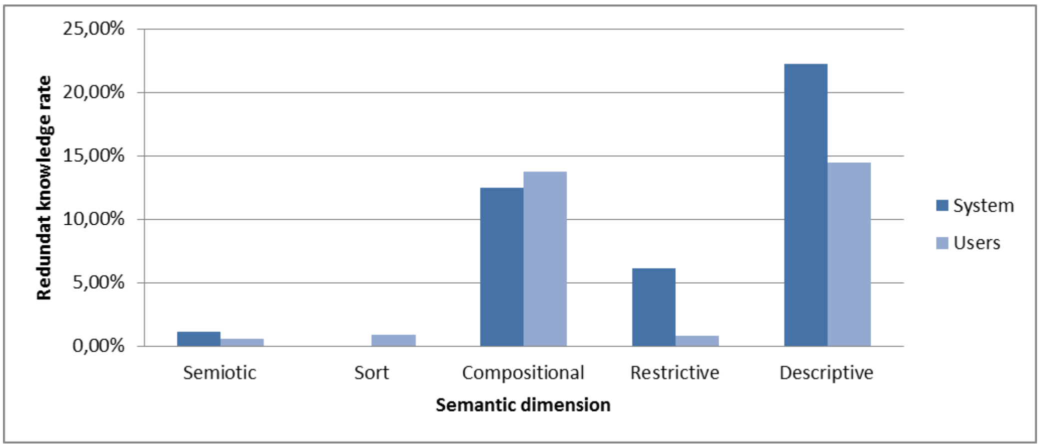

Figure 32 - Redundant knowledge per semantic dimension

5.2.4.2 *Results according to user experience levels.*

The following results present the comparison of the three measures according the user experience. In the first place, Figure 33 shows the hit rate comparison between system and users with different levels of experience.

If we focus on the first three semantic dimensions, users with more than ten years of experience in linguistic annotation achieve the best hit rate. The second best hit rate is achieved by the system in the two first dimensions (semiotic and sort) and by the users with medium experience in the compositional dimension. In these dimensions system tends to behave as the users with between one and two years of experience. In the case of the last two dimensions (restrictive and descriptive) results are better for system than for any group of experts.

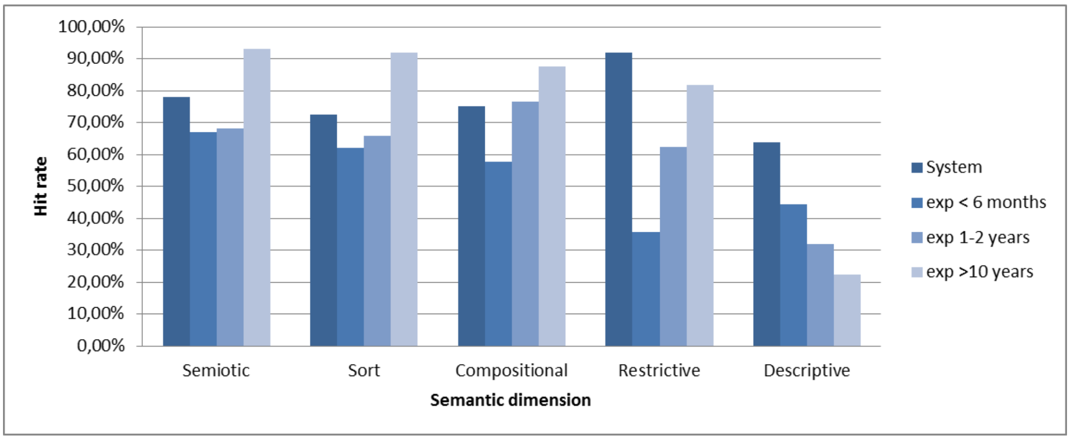

Figure 33 - Hit rate comparison between system and users with different levels of experience

In the second place, Figure 34 shows the buggy knowledge rate comparison between system and users with different levels of experience. In this case, system's results tend to be similar to those performed by the users with a medium level of experience, and are considerably high for the descriptive dimension.

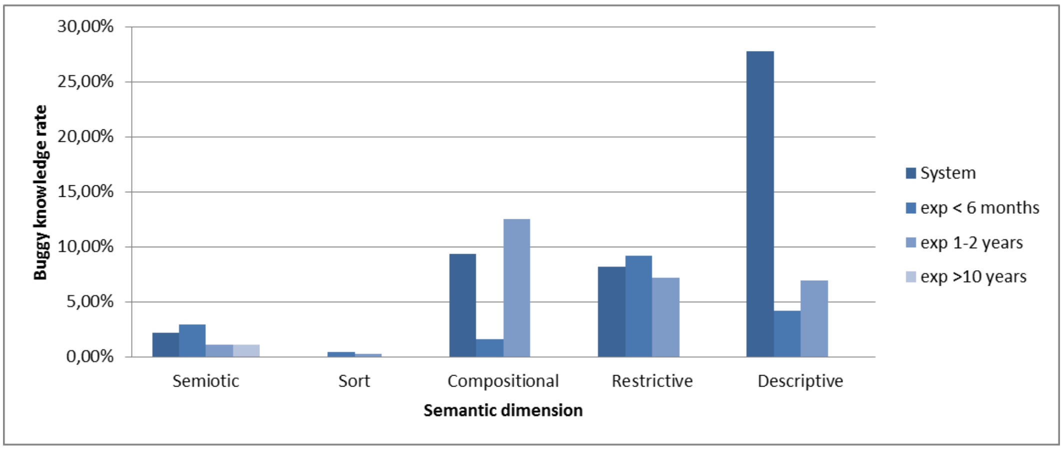

Figure 34 - Buggy knowledge rate comparison between system and users with different levels of experience

In the third place, Figure 35 shows the redundant rate comparison between system and users with different levels of experience. The highest rates of redundant knowledge occur in the compositional and descriptive dimensions, both for users and the system.

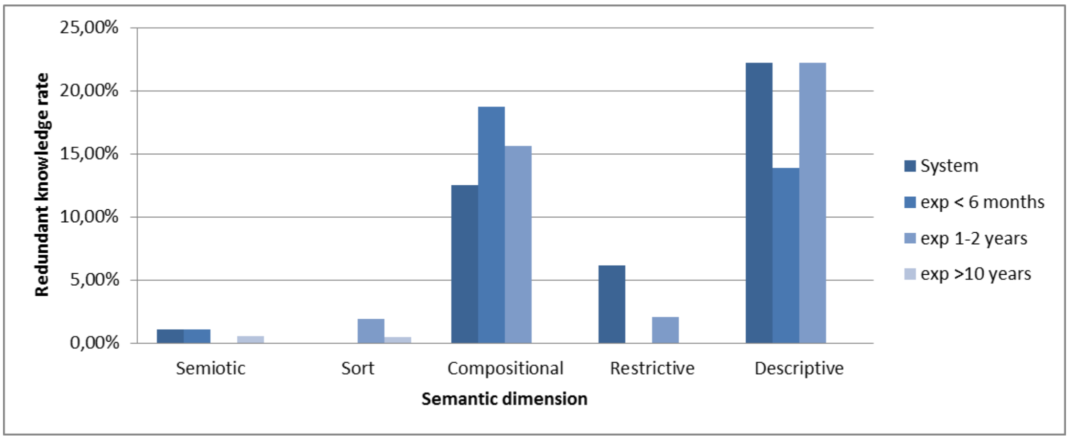

Figure 35 - Redundant knowledge rate comparison between system and users with different levels of experience

5.2.4.3 Results according to the language of text.

The following results present the comparison of the three measures depending on the language of the text.

In the first place, Figure 36 shows the hit rate comparison between system and users according to the language. For the semiotic, sort restrictive and descriptive dimensions, results are similar in both languages. Compositional dimension only get different results depending on the language, which achieves better hit rate in Spanish than in the English.

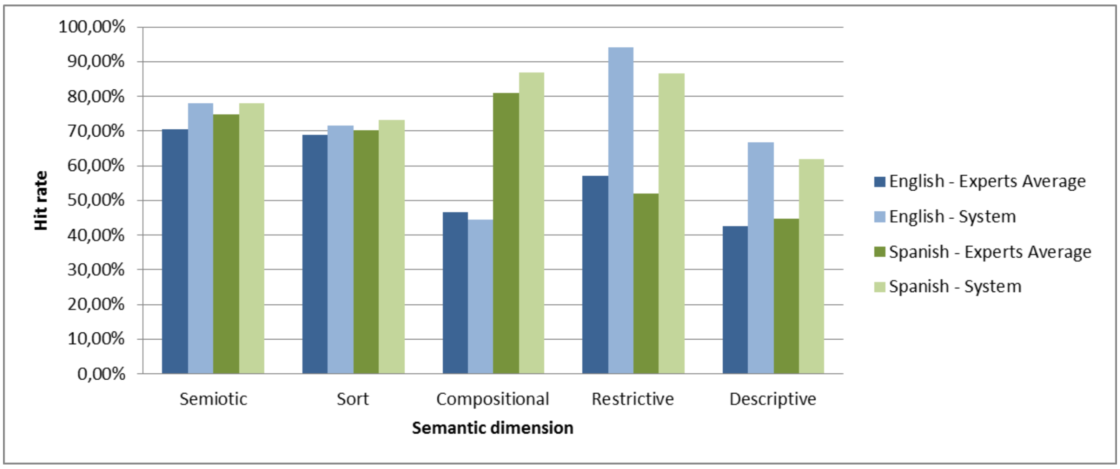

Figure 36 - Hit rate comparison between system and users according to the language

In the second place, Figure 37 shows the buggy knowledge rate comparison between system and users according to the language. The buggy knowledge rate is higher for Spanish in the compositional and restrictive dimensions and for the rest is higher in the English texts, especially in the descriptive dimension where the system gets the highest rate.

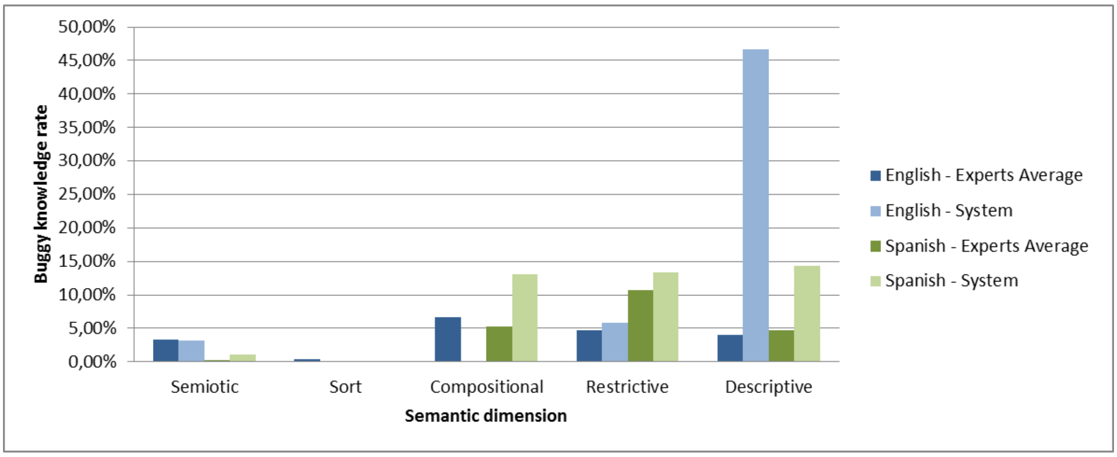

Figure 37 - Buggy knowledge rate comparison between system and experts according to the language

In the third place, Figure 38 shows the redundant rate comparison between system and users according to the language. In this figure, two results can be highlighted: on one hand, the compositional dimension obtains a higher redundant rate in Spanish and on the other hand, the descriptive dimension gets a higher redundant rate in English in the case of the system.

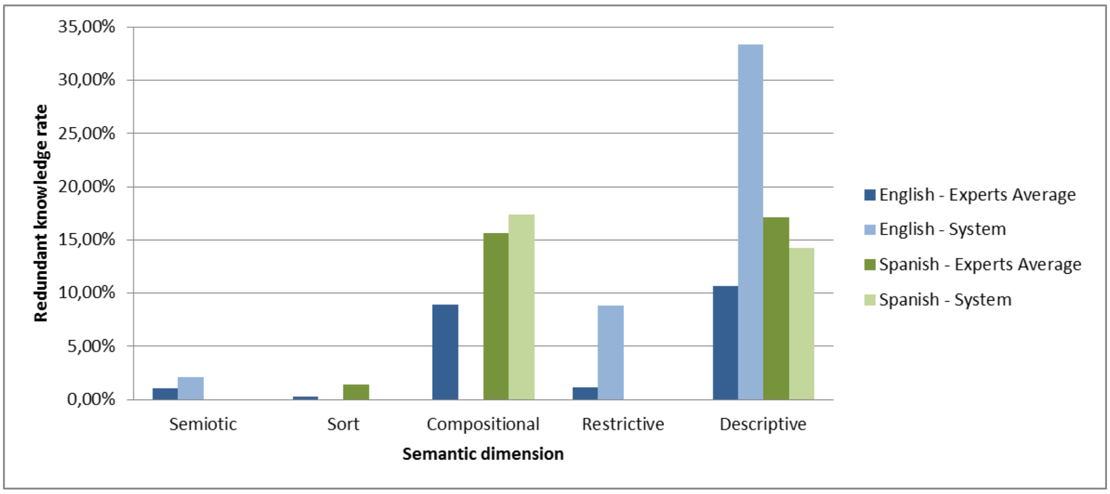

Figure 38 - Redundant knowledge rate comparison between system and experts according to the language

5.2.4.4 Results according to the type of text.

The following results present the comparison of the three measures depending on the type of text.

In the first place, Figure 39 shows the hit rate comparison between system and users according to the type of text. The better hit rates are achieved by the system for the science texts in the semiotic, sort and restrictive dimension. Economy texts get the better results in the compositional dimension and finally, fiction texts gets the better results in the case of the descriptive dimension.

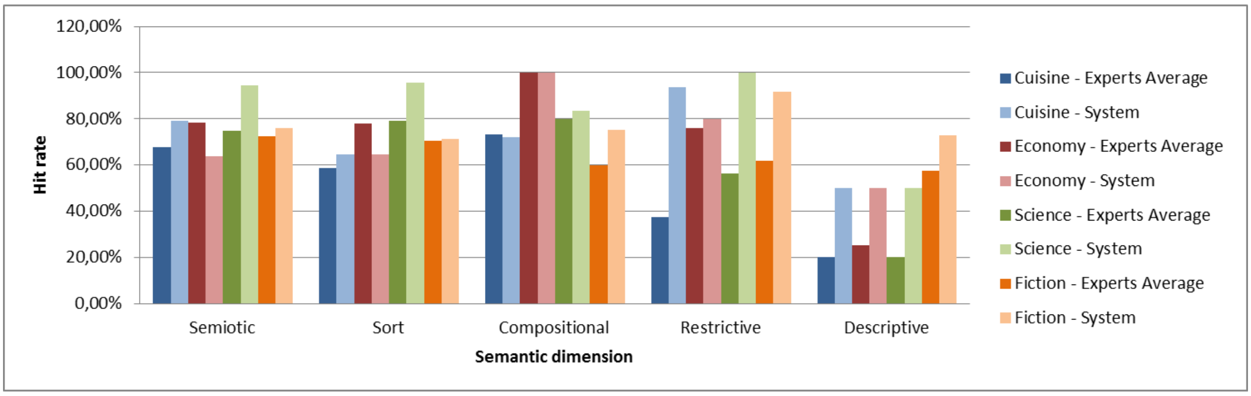

Figure 39 - Hit rate comparison between system and experts according to the type of text

In the second place, Figure 40 and Figure 41 show respectively, the buggy knowledge rate and the redundant knowledge rate comparison between system and users according to the type of text. Both figures show very high rates for the science and cuisine texts in the case of the system. In addition, economy and fiction texts obtain high buggy and redundant rates in descriptive and compositional dimensions, respectively.

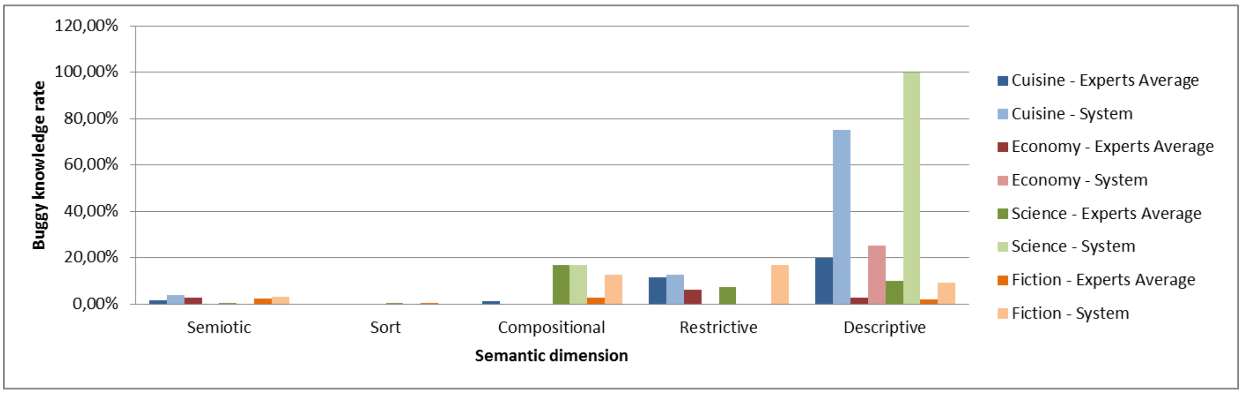

Figure 40 - Buggy knowledge rate comparison between system and experts according to the type of

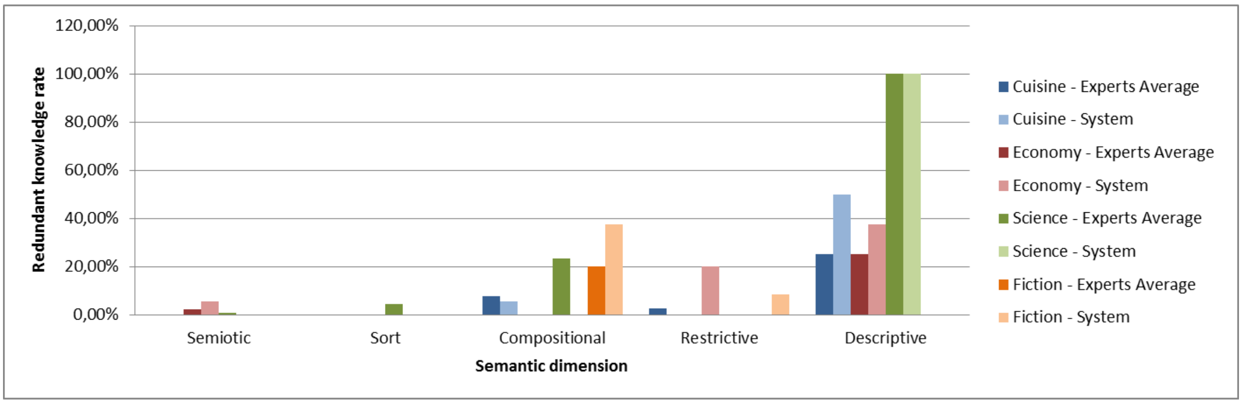

Figure 41 - Redundant knowledge rate comparison between system and experts according to the type of text

5.2.5 Discussion of the results obtained

First of all it is necessary to highlight that the method used to measure the quality of knowledge acquired by the ontology is a proposal of evaluation mechanism, since there are barely specific works in the area. In order to build the evaluation mechanism it has been defined a new methodology which is inspired by the evaluation of other approaches that include the human judgment in their assessments and the definition of a gold standard for comparing the quality of the results obtained by the acquisition system. The results obtained from the experiments described in the evaluation (Section 5.2.4) are analyzed and discussed below.

From a global point of view, experts get their best results in the first three dimensions (semiotic, sort and compositional), which are present in many other approaches, and for which they could have experience in annotating. Conversely, for descriptive and restrictive and dimensions, which are somewhat new, their results are not so good. This might be because both dimensions are part of the proposal of this work, so they do not have experience in annotating them (despite being very experienced in linguistic annotation).

On the other hand, system achieves better hit rate results than the users average in all dimensions, but the buggy and redundant knowledge rate obtained is too high in the new dimensions (restrictive and descriptive). The cause of these results obtained by the system can be attributed to the set of linguistic patterns that have been used for the recognition of ontological relationships. In the old dimensions (semiotic, sort and compositional) they were used NL patterns that have been developed by many expert during the years, as shown in the prior art, so the results are quite good. However, in the case of the novel dimensions (restrictive and descriptive), there are no patterns previously defined, nor related work in any studies in the state of the art, so it has been necessary to create whole new patterns. The patterns for those dimensions that exist for many years should be more complete and refined than those for the new dimensions, which are entirely new. Consequently, it would be appropriate to revise and expand the patterns applied to these new dimensions and review the acquisition functions in order to decrease the acquired buggy and redundant knowledge.

Regarding the analysis of results according to the user experience, it appears that the hit rate of the system approaches the hit rate of users with intermediate experience in the first three dimensions (semiotic, sort, compositional) and outperforms all user groups in the new dimensions (restrictive and descriptive). For attaining more significant results in this evaluation, it would be desirable to count on analysts with experience in annotating all dimensions (or, at least, increase the training phase up to several days).

6 Conclusions

Through this work it has been achieved the following impact: an ontology with more knowledge than the rest of ontologies defined in the state of the art, a method for calculating similarities between concepts that gets better results than those mentioned the state of the art and, finally, an ontology population mechanism which prevents the ontology to be populated manually.

In the first place, this work defines a new ontological model oriented to human-like interaction systems. This model is based on seven knowledge dimensions: semiotic, sort, compositional, essential, restrictive, descriptive and comparative. The first four dimensions are founded on the state of the art and the next three are part of the proposal of this work. In addition, although there are previous works that include the essential dimension, in this case is specifically designed to solve problems of human interaction, so its design is also new.

Secondly, this work defines a similarity measure for a multi-dimensional knowledge model of the ontology type, specifically an ontology aimed at supporting Human-Like Interaction. The proposed measure is based on five dimensions of ontological knowledge: sort, compositional, essential, restrictive and descriptive. The five of them are weighted and aggregated in order to obtain a global similarity measure. The equations applied for each dimension are general and can be used with other ontologies that observe any of these dimensions, yet observing all of them and aggregating their similarity result is here proposed for enhanced accuracy.

This solution presents another challenge, in the form of those weights calculation. In fact, when a person decides the similarity between concepts he unwittingly makes some dimensions prevail over the others. The criteria may be diverse, and this work has focused on studying the dependence of these weights on the nature of concepts, either in pairs (pair training method) or individually (feature training method), both described in Section 4.3. But this work also explores the influence of the past behavior of users who perform the concept pair evaluations (and ultimately, the user who owns a device or usually interacts with it). Following this line, a user-dependent training is proposed, and finally a hybrid one (merging feature and user benefits) is included too. All of them have been evaluated and compared in order to ascertain which one performs better, obtaining the best results for the pair-oriented training.

In order to evaluate the performance of the proposed similarity measure, its results were recorded and compared with those taken from human test subjects. This evaluation technique has been applied in several studies about similarity measures and is considered the gold standard. In the experimental phase, four training algorithms were developed according to different perspectives. Thus, this phase included a pair-oriented, a feature-oriented, a user-oriented and a hybrid experiment. In every case, the error rate was calculated with respect to the human subject assessments. The best results corresponded to the pair-oriented method which achieved an error rate of 18.5%. Since the implementation of this experiment is not realistic with large ontologies, a feature-oriented experiment was required despite slightly worsening the results from the previous experiment, concretely, producing an error rate of 20.2%. However, the feature-oriented experiment has the big advantage of being able to be applied easily to large ontologies.

Moreover, the user-oriented training aimed to adapt the weights to each subject in order to confirm the assumption that not every test subject assigns the same value to all dimensions. While this experiment had the highest error rate of all the algorithms (23.9%), as has been demonstrated, the error rate follows a decreasing trend line while, if training is not done, the error rate follows an asymptotic tendency. In addition to this, this experiment shows slightly better results than taking into account only the sort dimension (which has an average error rate of 24.1% and maximum 60.4%). For this reason, it can be concluded that the user-oriented experiment is able to adapt to each individual judgment (although this adaptation is very slow). Finally, the hybrid experiment combines the feature-oriented and the user-oriented training and, with an error rate of 21.2%, nevertheless manages to reduce the error of the user-oriented training, as well as balancing the error in the atypical cases common to the rest of the experiments.

The manual acquisition of new knowledge by an expert requires a great deal of resources and it would be desirable to develop an advanced mechanism to learn new concepts and relations. The third research line of this work focuses on the developing of an automatic mechanism to acquire ontological knowledge. The challenge is to attain that knowledge acquisition through human-like interaction with human subjects. Therefore, through the lifetime of the system, the knowledge bases will be enriched by interacting with the users. For this proposal, four ontology learning services are proposed and implemented to acquire new terms, concepts and relations and, finally, to unify concepts.

For evaluating the accomplishment of this third part, a novel evaluation methodology has been proposed, defined and carried out with a set of users with experience in linguistic corpus annotation. This mechanism is an evaluation proposal, since there has been no similar work in this area. The new methodology defined is inspired by the evaluation of other proposals in the area that include the human judgment. This mechanism is characterized by being clear and meaningful.

The results of this novel evaluation show that users get the best results in those dimensions already known by them (semiotic, sort and compositional) and for which there is already previous work in the state of the art. However, the results obtained by the users for the new dimensions (restrictive and descriptive) are not so fine since users do not have experience with them, despite being very experienced in linguistic annotation. In addition, results follow a common pattern depending on the user experience level.

In the case of the results obtained by the system, it achieves better hit rate results than the users average in all dimensions, but the buggy and redundant knowledge rate obtained is higher in the new dimensions than for the users. This can be explained because the system is able to offer and acquire many semantic relationships, but that may not be correct or may introduce redundant information defined already by other relations.

The cause of the results obtained by the system can be attributed to the set of linguistic patterns that have been used for the recognition of ontological relationships. In the old dimensions (semiotic, sort and compositional) they were used patterns that have been developed for a long time and is shown in the prior art, so the results are good. However, in the case of the novel dimensions, there are no patterns defined and refined by other studies in the state of the art, so it has been necessary to create new linguistic patterns. Thus, it can be concluded that would be appropriate to revise and expand the patterns used by this new dimensions and review the acquisition functions in order to decrease the acquired buggy and redundant knowledge. Besides, the proposed evaluation mechanisms provides tools to refine both the set of patterns and the acquisition functions, by analyzing the deviations between the gold standard and the system response.

7 Perspective for Future Research

In this section will be described the research lines left open after this work. This proposal next challenges can be distributed into three main issues: the knowledge model definition, the semantic similarity function and the automatic acquisition of knowledge. Besides, there are some noteworthy issues regarding the implementation approach.

Indeed, we should take in mind that success in applying this sort of knowledge model is linked to the efficiency of its implementation. No wonder, most of the systems that could take advantage of its virtues must provide responses in real time. Therefore, enhancing efficiency is a remarkable line of work, that could even affect the definition of the knowledge model. In (Calle et al, 2008), the authors proposed the inclusion of a seventh dimension aimed to improve efficiency when providing similarity services. The so named *comparative dimension* was set as a dimension derived from the other six as a mechanism for buffering most commonly compared pairs of concepts. The facts stored within this dimension are continuously being reviewed and recalculated offline, while are immediately available when required in running time. Furthermore, divergences found when reviewing can be profited to refine the here proposed weight-based mechanisms, so its parameters will be progressively refined to be suited to the specific domain in multi-user implantations (set for several anonymous users), or suited to each specific user in individual-user implantations (those set for a single user, or several identified users). In addition, such dimension in individual-user implantations can take profit from a User Model. That sort of knowledge model provide relatedness between registered users, so the knowledge learnt for a user can be to some extent applied to close-related others. On one hand, the User Model can support the Ontology in computing the weights vector for similarity calculation. On the other hand, the previously calculated similarities for related users can be used for providing a quick response when that fact is not available in the comparative dimension for current user, and could be applied for reviewing the buffered measures of similarity later on. In sum, this line can contribute to improve both efficiency and efficacy.

In a complementary way, and within the scope of improving efficiency, it should be stated that current implementation of the system is supported by Oracle Database, a Dialogue System implemented in Java, and a Natural Language processor based on OpenNLP. The efficiency of the DBMS is essential, since the knowledge bases are the cornerstone of the here proposed model. Though current implementation is adequate for the goals of this work, it could be improved by applying in-memory database management, such as for example the provided by TimesTen (which is also a product from Oracle Corp., entailing minimum changes on current implementation). That sort of technology is highly demanding in hardware resources, but boost efficiency minimizing response time when querying. The Dialogue and NL technologies should also be evolved, to improve response time and also for extending the reach of current implementation (which is of single-user type).

The latter line of work is related to the amount of knowledge stored in the database. The effectiveness of both semantic similarity function and the mechanisms of knowledge acquisition, and in fact of any other facility provided by this knowledge model, is closely related to the amount of stored knowledge. To definitely improve its performance involves aiming a common ontology model, supporting and supported by as many users as possible (ideally, a global worldwide implantation). Through internet it would be possible to populate ontological knowledge bases on the network using free tools and APIs available to users. In this way, it would be created a globally available online ontological knowledge base with which any other ontology could share information. In addition, it could be created mechanisms by which a private ontology would share only part of his knowledge to the global ontology.

That approach, however, entails negative effects, and to avoid or minimize them necessarily grabs our attention. The pursued 'knowledge flood', consequence of our efforts in extending the reach and globalizing the Model, brings contradictory facts, buggy knowledge, and certainty drop. The solution might be found exploring the application of reputation techniques to each different knowledge source, which affects the 'confidence degree' of each new fact (this concept is already in use in current model).

In another vein, though the Model presented in this work has proven state of the art efficacy for certain tasks, several improvements on the ontological dimensions have been found through the evaluation, and constitute an exciting line of work. Logically, most of the attention should be focused on the two completely new knowledge dimensions, since the others have been developed and refined by many other works, and show less space for improvements. The two dimensions we are talking about are the *restrictive* and the *descriptive* ones.

On one hand, the restrictive dimension could be evolved to define the roles of the objects in the restriction. Each constraint should take the role of the associated object the restriction (on which it can be done something, the subject affected, the circumstances, etc.). Therefore, it would be necessary to add the parameter "role". This development could affect the performance of the calculation of similarities, but would improve their results, as well as the other functions. For this reason it is an interesting line to explore

On the other hand, the descriptive dimension implemented does not distinguish between specific and generic concepts. Currently, all items stored in the knowledge base are treated as generic concepts. Having both types allow us to infer general knowledge from inferred, which respond to the experience of the system. Storing the two types of concepts is the best option since it would support much better the dialogue. When two people converse often make reference to specific concepts much more than generic concepts. However, adding specific concepts into the system supposes a huge increase in the volume of knowledge that could only be undertaken if is chosen a solution containing big data.

With respect to the second proposal of this work, the semantic similarity function, several improvements can be performed. Since the hybrid experiment manages to balance the results of the other experiments, an improved hybrid algorithm will be developed. In this algorithm the calculation of the weights of each iteration will be affected depending on the error produced in the feature experiment for the pair of concepts corresponding to that iteration. Refinement of similarities formulation is also an interesting line of work, especially in the semiotic dimension for reintroducing its influence in the global similarity calculation.

The performance of the training methods proposed is closely related to the available extent of knowledge. For this reason, it is defined the third proposal of this work, the automatic acquisition knowledge system which is currently being improved for increasing the quality and completeness of the ontological knowledge.

Regarding the mechanisms for automatic acquisition of ontological knowledge, once the learning system has been refined and stabilized, it will be studied to use a semi-automatic or automatic evaluation technique (Ovchinnikova & Kühnberger, 2006). In this automatic acquisition of linguistic knowledge, context plays a key role. For example, assuming that the system does not know the term 'gift', the following sentence is introduced in the acquisition system to explain its meaning and help the ontology to determine the concept associated to the new term:

‘A **gift** is the same as a **present’**

At this point, the system would acquire a new term associated to all concepts related to the term ‘present’:

- Concept 1: present, nowadays (the period of time that is happening now; any continuous stretch of time including the moment of speech) "that is enough for the present"; "he lives in the present with no thought of tomorrow".
- Concept 2: present (something presented as a gift) "his tie was a present from his wife".
- Concept 3: present, present tense (a verb tense that expresses actions or states at the time of speaking).

However, the acquisition system just would have to learn that the term 'gift' is associated with the Concept 2. Consequently, this decision generates a lot of buggy knowledge that needs to be cleaned later by the dissociation learning service.

In order to minimize the buggy knowledge acquired by the learning services that must be cleaned later it would be desirable to develop an ‘identification’ function between concepts. This function would enable to perform two operations:

- In the process of acquiring knowledge, either by dialogue or text, the new concepts and relationships are not learned directly. Instead, a temporary mini-ontology is created with all the knowledge acquired in the session (in whole dialogue, or the entire text). After the complete mini-ontology is learning at the end of the session. The concepts of the mini-ontology will attempt to be identified with those already present in the super-ontology (stable knowledge).

  In the cases where the identification is positive (above a certain threshold, which needs to be trained) knowledge is learned against that (those) concept (s) identified (s). When such identification is zero, they will be new concepts. When the identification is diffuse (similarity between zero and the threshold) they are learned as new concepts, but they maintain a certain similarity with the identified concepts (this relationship is stored in the comparative dimension, with that degree of similarity).

- Subsequently, automatic cleaning processes (identification and dissociation) are executed on the knowledge base. In the identification process, the pairs of concepts that have related to certain degree of similarity and those concepts for which the similarity may be increased (because they have learned new knowledge to one of them or both) are compared again. In case the similarity increases and exceeds the threshold, they are identified as a single concept.

This scheme has in mind the context, and since the buggy knowledge is not stored, it reduces the need for dissociation. To implement this mechanism, it would be helpful to have a record of learning. Since the amount of knowledge would be very large, it would be essential to deploy a big data solution.

Finally, we want to emphasize that this proposal is a supporting tool, not a system itself. So all the advances presented, and future ones in these lines, are completely useless without proper and successful applications of the here proposed technology. In such vein, is particularly relevant its application to human-computer interaction, for which this sort of models are usually set. But, specifically, we should highlight that its features and facilities are suited to human-like interaction (also known as Robot-Human or Natural Interaction), which is a fascinating line of research which is now experiencing a boost due the research in mobile appliances performed by big companies of that area. The use of an ontology model as the proposed in this work in those applications, would substantially improve the communication with the user, making it more human. On the other hand, the ontology would greatly benefit from such interaction, enabling the automatic acquisition of knowledge.

Another possible application of this ontology can be found in the speech analytics systems. The ontology defined in this work could improve the quality and completeness of the linguistic annotations performed in order to support the analysis of user interactions.

Finally, we are currently applying this tool to re-structuring massive data storages (Big Data), which is often of unstructured nature, which hardens building queries and endangers the successful use of the information. The operation of the here proposed Ontology Model, as described in Section 4.1, is essential for successfully mapping terms underlying the same sort of data, restoring links between storage units lost due to diversity, and improving the analytic processing (when grouping and aggregating) of semantically related data.

8 References

Ahmad, K. T. (2003). Corpus-based thesaurus construction for image retrieval in specialist domains. *Advances in Information Retrieval*, 76.

Aliabadi, P., Ahmadi, M., Salavati, S., & Esmaili, K. (2014). Towards Building KurdNet, the Kurdish WordNet. *Proceedings of the 7th International Global WordNet Conference.* Tartu, Estonia.

Altintas, E., Karsligil, E., & Coskun, V. (2005). A new semantic similarity measure evaluated in word sense disambiguation. *Procs. of the 15th NODALIDA conference.* Joensuu.

Azarova, I., Mitrofanova, O., Sinopalnikova, O., Yavorskaya, A., & Oparin, M. (2002). RussNet: Building a Lexical Database for the Russian Language. *Workshop on Wordnet Structures and Standardisation and How this affect Wordnet Applications and Evaluation*, (pp. 60-64). Las Palmas.

Baader, F., Lutz, C., Sturm, H., & Wolter, F. (2002). Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, 1-58.

Baroni, M. B. (2004). Using cooccurrence statistics & the web to discover synonyms in a technical language. *4th International Conference on Language Resources and Evaluation, vol 5*, (pp. 1725–1728).

Bateman, J. A., Hois, J., Ross, R., & Tenbrink, T. (2010). A linguistic ontology of space for natural language processing. *Artificial Intelligence, Volume 174, Issue 14*, 1027-1071.

Berland, M. C. (1999). Finding parts in very large corpora. *37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, (pp. 57-64). Maryland.

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. *37th Annual Meeting of the Association for Computational Linguistic*, (pp. 57-64). Maryland, USA.

Berners-Lee, T. (1994). *World Wide Web Consortium*. Retrieved from http://www.w3.org/

Berners-Lee, T., & Fischetti, M. (1999). *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web.* New York: HarperBusiness.

Bernstein, A., Kaufmann, E., Kiefer, C., & Bürki, C. (2005). *SimPack: A Generic Java Library for Similarity Measures in Ontologies.* Zurich.

Bisson, C., Nedellec, C., & Canamero, L. (2000). Designing clustering methods for ontology-building - The Mo'K Workbench. *Proceedings of the ECAI Ontology Learning Workshop.*

Bond, F., Isahara, H., Fujita, S., Uchimoto, K., Kuribayashi, T., & Kanzaki, K. (2009). Enhancing the Japanese WordNet. *7th Workshop on Asian Language Resources.* Singapore.

Bond, F., Tze Lim, L., Kong Tang, E., & Riza, H. (2014). The combined Wordnet Bahasa. *NUSA: Linguistic studies of languages in and around Indonesia 57*, 83–100.

Brickley, D., & Guha, R. V. (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. Retrieved from RDF Vocabulary Description Language 1.0: RDF Schema: http://www.w3.org/TR/rdf-schema/

Buitelaar, P. C. (2005). Ontology Learning from Text: Methods, Evaluation and Applications. *IOS Press*.

Calle, F. (2004). *Interacción Natural mediante procesamiento intencional: Modelo de Hilos en diálogos.* Madrid: Tesis doctoral.

Calle, F., Castro, E., & Cuadra, D. (2008). Ontological dimensions applied to Natural Interaction. *Proceeedings of the 2008 First International Workshop on Ontologies in Interactive Systems*, (pp. 91-96).

Caraballo, S. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. *37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*, (pp. 120-126). Maryland.

Chapman, R. L. (1992). *Roget International Thesaurus 5 .* HarperCollins.

Chaudhri, V., Farquhar, A., Fikes, R., Karp, P. D., & Rice, J. P. (1998). OKBC: A programmatic foundation for knowledge base interoperability. *AAAI '98/IAAI '98 Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence* (pp. 600-607). Madison, Wisconsin, USA: American Association for Artificial Intelligence.

Chih-Yao, L., Yu-Yun, C., Shu-Kai, H., Jia-Fei, H., & Chu-Ren, H. (n.d.). A Wiki that Helps Quicken the Development of Chinese Wordnet. *ASIALEX 8th International Conference.* Bali, Indonesia.

Ciaramita, M. G. (2005). Unsupervised Learning of Semantic Relations between Concepts of Molecular Biology Ontology. *International Joint Conference on Artificial Intelligence.*

Cimiano, P. (2006). Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. New York: Springer-Verlag New York, Inc.

Cimiano, P. H. (2004). Towards the self-annotating web. *13th International Conference on World Wide Web*, (pp. 462-471). New York, USA.

Cimiano, P., & Wenderoth, J. (2007). Automatic acquisition of ranked qualia structures from the web. *Annual Meeting of the Association for Computational Linguistics*, (pp. 888-895). Prague.

Conesa, J., & Olivé, A. (2004). A General Method for Pruning OWL Ontologies. *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2004.* Agia Napa, Cyprus: Springer-Verlag.

Cook, D. (2008). MLSN: A multi-lingual semantic network. *14th Annual Meeting of the Association.*

Couto, F., Silva, M., & Coutinho, P. (2007). Measuring semantic similarity between gene ontology terms. *Data & Knowledge Engineering, 61(1)*, 137 – 152.

Cruse, D. (1986). *Lexical Semantics.* New York: Cambridge University Press.

Cycorp. (2013, 12 12). *OpenCyc*. Retrieved from OpenCyc: http://www.cyc.com/platform/opencyc

Dagan, I., Glickman, O., & Magnini, B. (2005). *The PASCAL Recognising Textual Entailment Challenge.*

Damerau, F. (1993). Evaluating domain-oriented multiword terms from texts. *Information Processing and Management 29(4)*, 433–447.

Dean, M., & Schreiber, M. (2004, February 10). *OWL Web Ontology Language Reference.* Retrieved from OWL Web Ontology Language Reference: http://www.w3.org/TR/2004/REC-owl-ref-20040210/

Department of Genetics, Stanford University School of Medicine. (2000). Gene ontology: tool for the unification of biology. *Nature genetics Vol. 25, No. 1*, 25-29.

Faatz, A. S. (2002). Ontology Enrichment with texts from the WWW. *Semantic Web Mining 2nd Workshop at ECML/PKDD-2002.* Helsinki, Finland.

Faure, D., & Nedellec, C. (1998). A corpus-based conceptual clustering method for verb frames and ontology. *Proceedings of the LREC Workshop on Adapting lexical and corpus resources to sublanguages and applications.* P. Velardi.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* The MIT Press: Cambridge, UK.

Fernández, M., Gómez-Pérez, A., & Juristo, N. (1997). METHONTOLOGY: From Ontological Art Toward Ontological Engineering. *Spring Symposium Series on Ontological Engineering*, 33-40.

Fortuna, B. M. (2005). Visualization of Text Document Corpus. *ACAI 2005 Summer School.*

Frantzi, K. A. (2000). Automatic recognition of multi-word terms: The c-value/nc-value method. *International Journal on Digital Libraries 3(2)*, 115–130.

García-Castro, R., Gómez-Pérez, A., & Muñoz-García, O. (2008). The Semantic Web Framework: a component-based framework for the development of Semantic Web applications. *7th Internacional Workshop on Web Semantics (WebS 2008).* Turin, Italia.

Garrett, M. F. (1982). Production of Speech: Observations from Normal and Pathological Language Use. *Normality and Pathology in Cognitive Functions*, A. Ellis.

Gee, J. (1999). *Introduction to Discourse Analysis.* Routledge.

Girju, R. B. (2006). Automatic discovery of part-whole relations. *Computational Linguistics 32 (1)*, 83-135.

Global WordNet Association. (2000). *Global WordNet Association*. Retrieved from Global WordNet Association: http://globalwordnet.org/

Gómez-Pérez, A., & Manzano-Macho, D. (n.d.). *A survey of ontology learning methods and tecniques*. Retrieved from Onto-web IST Project, Deliverable 1.5: http://www.ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable%201.5

González Agirre, A., Laparra, E., & Rigau, G. (2012). Multilingual Central Repository version 3.0: upgrading a very large lexical knowledge base. *International Conference on Language Resources and Evaluation*, (pp. 2525-2529). Istanbul, Turkey.

Gretz, S., Itai, A., MacWhinney, B., Nir, B., & Wintner, S. (2015). Parsing Hebrew CHILDES Transcripts. *Language Resources and Evaluation, 49(1)*, 107-145.

Griesel, M., & Bosch, S. (2014). Taking stock of the African Wordnets project: 5 years of development. *Global Wordnet Conference*, (pp. 148-153). Tartu, Estonia.

Gross, D., Fischer, U., & Miller, G. (1989). The organization of adjective meanings. *Journal of memory and Language, 28*, 92-106.

Guarino, N. (1998). Formal Ontology in Information Systems. *Proceedings of the 1st International Conference.* Trento, Italy: IOS Press.

Gupta, K. A. (2002). An Architecture for engineering sublanguage WordNets. *First International Conference On Global WordNet*, (pp. 207–215). Mysore.

Hamp, B., & Feldweg, H. (1997). GermaNet - a Lexical-Semantic Net for German. *Proceedings of ACL workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications.* Madrid.

Harris, Z. (1968). *Mathematical Structures of Language. John Wiley & Sons.* Chichester.

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. *14th Conference on Computational linguistics–vol. 2* (pp. 539–545). Nantes, France: Morgan Kaufmann Publishers.

Hearst, M. (1998). WordNet: An electronic lexical database and some of its applications. *Automated Discovery of WordNet Relations*.

Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., et al. (2000). *The Ontology Inference Layer OIL.* Amsterdam: Technical Report IR-479.

Hotho, A., & Staab, S. (2004). Clustering concept hierarchies from text. *Proceedings of LREC.*

Informatics, S. C. (2013). *Protégé*. Retrieved from Protégé: http://protege.stanford.edu/

Inkpen, D. (2007). Semantic similarity knowledge and its applications. *STUDIA UNIV. BABES-BOLYAI, INFORMATICA, Volume LII*, 11-22.

Ishiguro, H. (1990). *Leibniz's Philosophy of Logic and Language.* Cambridge: Cambridge University Press.

Jha, S., Narayan, D., Pande, P., & Bhattacharyya, P. (2001). A WordNet for Hindi. *International Workshop on Lexical Resources in Natural Language Processing.* Hyderabad, India.

Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. *In International Conference Research on Computational Linguistics.* Taiwan.

Kahusk, N., Orav, H., & Vare, K. (2012). Cross-linking Experience of Estonian WordNet. *Human Language Technologies – The Baltic Perspective: The Fifth International Conference on Human Language Technologies* (pp. 96 - 102). Tartu, Estonia: IOS Press.

Karp, P., Chaudhri, V. K., & Thomere, J. (1999). *XOL: An XML-based ontology exchange language*. Retrieved from XOL: An XML-based ontology exchange language: http://www.ai.sri.com/pkarp/xol/xol.html

Kavalec, M. M. (2004). Discovery of Lexical Entries for Non-taxonomic Relations in Ontology Learning. *SOFSEM.*

Keyvan, F., Borjian, H., Kasheff, M., & Fellbaum, C. (2006). Developing PersiaNet: The Persian WordNet. *Proceedings of the Third Global WordNet Meeting.* Jeju Island, Korea.

Kilgariff, A. (2007). Googleology is bad science. *Computational Linguistics 3 (1)*, 147–151.

Knight, K., & Luk, S. (1994). Building a Large Knowledge Base for Machine Translation. *Proceedings of the American Association of Artificial Intelligence Conference AAAI-94.* Seattle, WA.

Koeva, S., Leseva, S., Rizov, B., Tarpomanova, E., Dimitrova, T., Kukova, H., et al. (2011). Design and Development of the Bulgarian Sense-Annotated Corpus. *Actas del 3 Congreso Internacional de Lingüística de Corpus. Tecnologias de la Información y las Comunicaciones: Presente y Futuro en el Análisis de Corpus*, (pp. 143-150). Valencia.

Kozakov, L. P. (2004). Glossary extraction and utilization in the information search and delivery system for IBM Technical Support. *IBM System Journal 43(3)*.

Krauthammer, M., & Nenadic, G. (2004). Term identification in the biomedical literature. *Journal of Biomedical Informatics 37*, 512–526.

Kulkarni, M., Dangarikar, C., Kulkarni, I., Nanda, A., & Bhattacharyya, P. (2010). Introducing sanskrit wordnet. *Proceedings of the Global Wordnet Conference 2010.*

Laroche, N., Urdang, L., & Rodale, J. I. (1986). *The Synonym Finder REVISED Edition.* Grand Central Pub Mass Market.

Lassila, O., & McGuinness, D. (2001). The role of frame-based representation on the semantic web. *Linköping Electronic Articles in Computer and Information Science, vol 6*.

Lassila, O., Swick, R. R., Wide, W., & Consortium, W. (1998). *Resource Description Framework (RDF) Model and Syntax Specification*. Retrieved from Resource Description Framework (RDF) Model and Syntax Specification: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Lenat, D., & Guha, R. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady vol 10*, 707-710.

Lin, D. P. (2001). Induction of semantic classes from natural language text. *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, (pp. 317–322).

Lin, D. P. (2002). Concept discovery from text. *International Conference on Computational Linguistics*, (pp. 577–583).

Lindén, K., & Carlson, L. (2010). FinnWordNet – WordNet på finska via översättning. *LexicoNordica – Nordic Journal of Lexicography*, 119–140.

Lord, P. W., Stevens, R. D., Brass, A., & Goble, C. A. (2003). Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. *Bioinformatics*, 1275-1283.

Luke, S., & Heflin, J. (2000). *SHOE 1.01. Proposed Specification*. Retrieved from SHOE 1.01. Proposed Specification: http://www.cs.umd.edu/projects/plus/SHOE/spec.html

Marrafa, P., Amaro, R., Chaves, R., Lourosa, S., Martins, C., & Mendes, S. (2005). WordNet.PT – Uma Rede Léxico-conceptual do Português on-line. *XXI Encontro da Associação Portuguesa de Linguística.* Porto, Portuga.

Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., et al. (2005). Searching for Common Sense: Populating Cyc from the Web. *20th National Conference on Artificial Intelligence.* Pittsburgh, PA.

Maziarz, M., Piasecki, M., & Szpakowicz, S. (2012). Approaching plWordNet 2.0. Proceedings of the 6th Global Wordnet Conference. Japan.

Miháltz, M., Hatvani, C., Kuti, J., Szarvas, G., Csirik, J., Prószéky, G., et al. (2008). Methods and Results of the Hungarian WordNet Project. *Proceedings of the Fourth Global WordNet Conference*, (pp. 310-320).

Miller, G. (1995). WordNet: A Lexical Database for English. *Communications of the ACM vol 38 ,No. 11*, 39-41.

Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 1-28.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, 235-244.

MIT, ERCIM, Keio, Beihang. (2013). *W3C DATA ACTIVITY Building the Web of Data*. Retrieved from W3C DATA ACTIVITY Building the Web of Data: http://www.w3.org/2013/data/

Mizoguchi, R., Van Welkenhuysen, J., & Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge. *In Towards Very Large Knowledge Bases* , 46-57.

Morato, J., Marzal, M. Á., Lloréns, J., & Moreiro, J. (2004). WordNet Applications. *Proc. of the 2nd Int. Conf. Global WordNet.* Brno, Czech Republic.

Moreno, A., Riaño, D., Isern, D., Bocio, J., Sánchez, D., & Jiménez, L. (2004). Knowledge exploitation from the web. *Fifth International Conference on Practical Aspects of Knowledge Management*, (pp. 175–185). Vienna, Austria.

Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., & Lutz, C. (2009). *OWL 2 Web Ontology Language: Profiles.* Retrieved from OWL 2 Web Ontology Language: Profiles: http://www.w3.org/TR/owl2-profiles

Musen, M. (2015). The protégé project: a look back and a look forward. *AI Matters Volume 1 Issue 4*, 4-12.

Navigli, R., & Pontezzo, S. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence, 193, Elsevier*, 217-250.

NeOn Foundation. (2013). *NeOn Toolkit*. Retrieved from NeOn Toolkit: http://neon-toolkit.org/wiki/Main_Page

Nevěřilová, Z. (2009). Exploring and Extending Czech WordNet and VerbaLex. *Proceedings of the RASLAN Workshop 2009*, (pp. 69-82). Masaryk University.

North-West University, South Africa. (2013). *Afrikaanse Wordnet*. Retrieved from Afrikaanse Wordnet: http://www.nwu.ac.za/af/content/nwu-potchefstroomkampus-ctext-11

O'Hara, T., Salay, N., Witbrock, M., Schneider, D., Aldag, B., Bertolo, S., et al. (2003). Inducing criteria for mass noun lexical mappings using the Cyc Knowledge Base and its Extension to WordNet. *5th International Workshop on Computational Semantics.* Tilburg.

Oliveira, H., & Gomes, P. (2010). Onto.PT: Automatic Construction of a Lexical Ontology for Portuguese. *Proceedings of 5th European Starting AI Researcher Symposium* (pp. 199-211). IOS Press.

Ovchinnikova, E., & Kühnberger, K. (2006). Aspects of Automatic Ontology Extension: Adapting and Regeneralizing Dynamic Updates. *AOW '06 Proceedings of the second Australasian workshop on Advances in ontologies* (pp. 52-60). Darlinghurst, Australia: Australian Computer Society, Inc.

Pagels, M. (2000). *DAML- The DARPA Agent Markup Language Homepage*. Retrieved from DAML- The DARPA Agent Markup Language Homepage: http://www.daml.org/

Pantel, P., & Pennacchiotti, M. (2006). Espresso: leveraging generic patterns for automatically harvesting semantic relations. *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, (pp. 113-120). Sydney, Australia.

Pasca, M. V. (2007). The role of documents vs. queries in extracting class attributes from text. *Sixteenth Conference on Information and Knowledge Management*, (pp. 485-494). Lisboa, Portugal.

Pease, A., Niles, I., & Li, J. (2002). The Suggested Upper Merged Ontology: A Large Ontology for the Semantic Web and its Applications. *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web.*

Pedersen, T., Patwardhan, S., & Michelizzi, J. (2004). WordNet:: Similarity measuring the relatedness of concepts. *Demonstration Papers at HLT-NAACL 2004* (pp. 38-41). Boston, Massachusetts, USA: Association for Computational Linguistics.

Pianta, E., Bentivogli, L., & Girardi, C. (2002). MultiWordNet: developing an aligned multilingual database. *Proceedings of the First International Conference on Global WordNet*, (pp. 293-302). Mysore, India.

Pinto, H., Staab, S., & Tempich, C. (2004). DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. *Proceedings of the 16th European Conference on Artificial Intelligence* (pp. 393-397). Amsterdam: IOS Press.

Pivk, A. C. (2007). Transforming arbitrary tables into logical from with TARTAR. *Data and Knowledge Engineering 60 (3)*, 567–595.

Pushpak, B. (2010). IndoWordnet. *Proceedings of the International Conference on Language Resources and Evaluation*, (pp. 3785-3792). Valletta, Malta.

Quentin Pradet, G., & Baguenier Desormeaux, J. (2014). WoNeF, an improved, expanded and evaluated automatic French translation of WordNet. *Proceedings of the Seventh Global Wordnet Conference*, (pp. 32-39).

Rada, R., Mili, H., Bicknell, E., & Blettner, M. (1989). Development and application of a metric on semantic nets. *IEEE Trans. on Systems, Man and Cybernetics*, (pp. 17-30).

Raffaelli, I., Tadić, M., Bekavac, B., & Agić, Ž. (2008). Building Croatian WordNet. Proceedings of the 4th Global WordNet Conference. (pp. 349-359). Szeged: Global WordNet Association.

Ramesh Ram, N., & Namrata Mahender, C. (2014). Marathi WordNet Development. *International Journal of Engineering & Computer Science Vol. 3*, 7622-7624.

Ravi, S., & Pasca, M. (2008). Using structured text for large scale attribute extraction. *17th Conference on Information and Knowledge Management*, (pp. 1183-1192).

Reisinger, J., & Pasca, M. (2009). Low-cost supervision for multiple-source attribute extraction. *10th International Conference on Intelligent Text Processing and Computational Linguistics*, (pp. 382-393).

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. *IJCAI'95 Proceedings of the 14th international joint conference on Artificial intelligence* (pp. 448-453). San Francisco, USA: Morgan Kaufmann Publishers Inc.

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal Of Artificial Intelligence Research*, 95-130.

Rodríguez, H., Farwell, D., Farreres, J., Bertran, M., Alkhalifa, M., Martí, M., et al. (2008). Arabic WordNet: Current State and Future Extensions. *Proceedings of the Fourth International GlobalWordNet Conference – GWC 2008*, (pp. 22-25). Szeged, Hungary.

Ruiz-Casado, M., Alfonseca, E., & Castells, P. (2007). Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data and Knowledge Engineering 61 (3)*, 484-499.

Salton, G. W. (1975). A vector space model for automatic indexing. *Communications of the ACM 18(11)*, (pp. 613–620).

Sánchez, D. (2010). A methodology to learn ontological attributes from the Web. *Data Knowledge Engineering*, 573-597.

Sánchez, D. I. (2011). Automatic extraction of acronym definitions from the Web. *Application Intelligence*.

Sánchez, D. M. (2008). Learning non-taxonomic relationships from web documents for domain ontology construction. *Data and knowledge Engineering, Elsevier 63(3)*, 600-623.

Sánchez, D. M. (2008). Pattern-based automatic taxonomy learning from the Web. *AI Community 21*, 27-48.

Sánchez, D. M.-T. (2012). Learning relation axioms from text: an automatic Web-based approach. *Expert Systems Applications 39*, 5792–5805.

Sanderson, M. C. (1999). Deriving concept hierarchies from text. *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 206-213). Berkeley,California, USA.

Scannell, K. (2007). *Irish Language Semantic Network*. Retrieved from Irish Language Semantic Network: http://borel.slu.edu/lsg/index-en.html

Schickel-Zuber, V. (2007). OSS: a semantic similarity function based on hierarchical ontologies. *IJCAI'07 Proceedings of the 20th international joint conference on*

Artifical intelligence (pp. 551-556). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Schutz, A. B. (2005). RelExt: A Tool for Relation Extraction from Text in Ontology Extension. *4th International Semantic Web Conference.*

Sclano, F. V. (2007). TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities. *9th Conf. on Terminology and Artificial Intelligence.* Sophia Antinopolis.

Shamsfard, M., & Barforoush, A. A. (2003). The state of the art in ontology learning: a framework for comparison. *Knowl. Eng. Rev. 18(4)*, 293–316.

Sicilia, M., García, E., Sánchez, S., & Rodríguez, E. (2004). On Integrating Learning Object Metadata inside the OpenCyc Knowledge Base. *Fourth IEEE International Conference on Advanced Learning Technologies*, (pp. 900-901).

Sigel, A. (2006). From traditional Knowledge Organization Systems (authority files, classifications, thesauri) towards ontologies on the web. *9th International Conference of the International Society for Knowledge Organization* (pp. 3-53). Vienna, Austria: E-prints in Library and Information Science.

Suárez-Figueroa, M., Gómez-Pérez, A., & Fernández-López, M. (2012). The NeOn Methodology for Ontology Engineering. *Ontology Engineering in a Networked World*, 9-34.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A Core of Semantic Knowledge. *16th International Conference on World Wide Web* (pp. 697-706). Banff, Alberta, Canada: ACM.

Sure, Y., Akkermans, H., Broekstra, J., Davies, J., Ding, Y., Duke, A., et al. (2003). On-To-Knowledge: Semantic Web-Enabled Knowledge Management. In N. Zhong, J. Liu, & Y. Yao, *Web Intelligence* (pp. 277-300). Springer Berlin Heidelberg.

Sussna, M. (1993). Word sense disambiguation for free-text indexing using a massive semantic network. *Proceedings of the second international conference on Information and knowledge management* (pp. 67-74). Washington, DC, USA: ACM.

The Apache Software Foundation. (2010). *Apache OpenNLP*. Retrieved from Apache OpenNLP: https://opennlp.apache.org/

The Eclipse Foundation. (2015). *Eclipse*. Retrieved from Eclipse: https://eclipse.org/downloads/

The Nederlandse TaalUnie and the Free University of Amsterdam. (n.d.). *Open Dutch WordNet*. Retrieved from Open Dutch WordNet: http://wordpress.let.vupr.nl/odwn/

Thiyagarajan, S., Arulmozi, S., & Rajendran, S. (2002). Tamil WordNet. *First Global WordNet Conference, CIIL.* Mysore.

TopQuadrant. (2001). *TopBraid Composer*. Retrieved from TopBraid Composer: http://www.topquadrant.com/tools/ide-topbraid-composer-maestro-edition/

TopQuadrant. (2013). *TopBraid Composer*. Retrieved from TopBraid Composer: http://www.topquadrant.com/products/TB_Composer.html

Touretzky, D. (1986). *The mathematics of inheritance systems.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Trap-Jensen, L., Lorentzen, H., Hartvig Sørensen, N., Sandford Pedersen, B., Asmussen, J., & Nimb, S. (2009). DanNet: the challenge of compiling a wordnet for Danish by reusing a monolingual dictionary. *Language Resources and Evaluation 43(3)*, 269-299.

Tufis, D., Cristea, D., & Stamou, S. (2004). BalkaNet: Aims, Methods, Results and Perspectives. A General Overview. *Romanian Journal of Information, Science and Technology*, 9-43.

Tversky, A. (1977). Features of Similarity. *Psycological Review, 84*, 327-352.

Van Heijst, G., Schreiber, A. T., & Wielinga, B. J. (1997). Using explicit ontologies for KBS development. *International Journal of Human-Computer Studies, 46(2-3)*, 183–292.

Veale, T., & Hao, Y. (2007). Comprehending and generating apt metaphors: a web-driven, case-based approach to figurative language. *AAAI*, (pp. 1471-1476).

Vetulani, Z., Kubis, M., & Obrębski, T. (2010). PolNet – Polish WordNet: data and tools. *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, (pp. 3793–3797). Valletta, Malta.

Vlora University. (2008). *On the current state of Albanet and related applications.* Vlora, Albania: Vlora University.

Vossen, P. (1998). *EuroWordNet: A Multilingual Database with Lexical Semantic Networks.* Norwell, MA, USA: Kluwer Academic Publishers.

Weichselbraun, A. W. (2009). Discovery and evaluation of non-taxonomic relations in domain ontologies. *International Journal of Metadata, Semantics and Ontologies 4 (3)*, 212–222.

Welgama, V., Herath, D. L., Liyanage, C., Udalamatta, N., Weerasinghe, R., & Jayawardana, T. (2011). Towards a Sinhala Wordnet. *Conference on Human Language Technology for Development*, (pp. 39-43). Alexandria, Egypt.

Wong, K. (2008). *RiTa.WordNet: a WordNet library for Java/Processing*. Retrieved from RiTa.WordNet: a WordNet library for Java/Processing: https://rednoise.org/rita/

Wu, Z., & Palmer, M. (1994). Verb semantics and lexical selection. *ACL'94 Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133-138). Stroudsburg, USA: Association for Computational Linguistics.

Yarowsky, D. (1992). Word-sense disambiguation using statistical models of roget's categories trained on large corpora. *COLING 1992.* Nantes.

Yoon, A., Hwang, S., Lee, E., & Kwon, H.-C. (2009). Construction of Korean Wordnet. *Korean Institute of Information Scientists and Engineers : software and application 36(1)*, 92-108.

9 Appendix

WordNet Name	Languages	Reference
Afrikaans WordNet	Afrikaan	(North-West University, South Africa, 2013)
AlbaNet	Albanian	(Vlora University, 2008)
Arabic WordNet	Arabic	(Rodríguez, et al., 2008)
OpenMultilingual WordNet	Multilingual: Arabic, English, Malaysian, Indonesian, Finnish, Hebrew, Japanese, Persian, Thai and French	(González Agirre, Laparra, & Rigau, 2012)
WordNet Bahasa	Multilingual: Malaysian and Indonesian	(Bond, Tze Lim, Kong Tang, & Riza, 2014)
African WordNet	Multilingual: Bantu languages	(Griesel & Bosch, 2014)
Multilingual Central Repository 3.0	Multilingual: English, Spanish, Catalan, Basque and Italian	(González Agirre, Laparra, & Rigau, 2012)
BulNet	Bulgarian	(Koeva, et al., 2011)
BalkaNet	Multilingual: Bulgarian, Czech, Greek, Romanian, Serbian and Turkish	(Tufis, Cristea, & Stamou, 2004)
Academia Sinica Bilingual Ontological Wordnet	Chinese	(Chih-Yao, Yu-Yun, Shu-Kai, Jia-Fei, & Chu-Ren)
Croatian WordNet (CroWN)	Croatian	(Raffaelli, Tadić, Bekavac, & Agić, 2008)
Czech WordNet	Czech	(Nevěřilová, 2009)
DanNet	Danish	(Trap-Jensen, Lorentzen, Hartvig Sørensen, Sandford Pedersen, Asmussen, & Nimb, 2009)
Combinatorial and Relational Network as Toolkit for Dutch Language Technology (Cornetto)	Dutch	(The Nederlandse TaalUnie and the Free University of Amsterdam)
WordNet	English	(Fellbaum, 1998)
EuroWordNet English	English	(Vossen, 1998)
Estonian Wordnet (EstWN)	Estonian	(Kahusk, Orav, & Vare, 2012)
FinnWordNet	Finnish	(Lindén & Carlson, 2010)
WOLF	French	(Quentin Pradet & Baguenier Desormeaux, 2014)
WoNeF	French	(Quentin Pradet & Baguenier Desormeaux, 2014)
GermaNet	German	(Hamp & Feldweg, 1997)
Hebrew WordNet	Hebrew	(Gretz, Itai, MacWhinney, Nir, & Wintner, 2015)
Hindi WordNet	Hindi	(Jha, Narayan, Pande, &

		Bhattacharyya, 2001)
Hungarian WordNet (HuWN)	Hungarian	(Miháltz, et al., 2008)
MultiWordNet	Multilingual: Italian, Spanish, Portuguese, Hebrew, Romanian and Latin	(Pianta, Bentivogli, & Girardi, 2002)
Líonra Séimeantach na Gaeilge (LSG) Irish Language Semantic Network	Irish	(Scannell, 2007)
Japanese WordNet	Japanese	(Bond, Isahara, Fujita, Uchimoto, Kuribayashi, & Kanzaki, 2009)
Multi-Lingual Semantic Network project	Multilingual: Japanese, Chinese and German	(Cook, 2008)
IndoWordNet	Multilingual: Hindi, Assamese, Bengali, Bodo, Gujarati, Kannada, Kashmiri, Konkani, Malayalam, Meitei, Marathi, Nepali, Sanskrit, Tamil, Telugu, Punjabi, Urdu and Oriya.	(Pushpak, 2010)
KorLex (Korean WordNet)	Korean	(Yoon, Hwang, Lee, & Kwon, 2009)
KurdNet	Kurdish	(Aliabadi, Ahmadi, Salavati, & Esmaili, 2014)
Marathi WordNet	Marathi	(Ramesh Ram & Namrata Mahender, 2014)
PersiaNet	Persian	(Keyvan, Borjian, Kasheff, & Fellbaum, 2006)
FarsNet	Persian	(Shamsfard & Barforoush, 2003)
PlWordNet (Slowosiec)	Polish	(Maziarz, Piasecki, & Szpakowicz, 2012)
PolNet	Polish	(Vetulani, Kubis, & Obrębski, 2010)
Onto.PT – Portuguese WordNet	Portuguese	(Oliveira & Gomes, 2010)
WordNet.PT – Portuguese WordNet	Portuguese	(Marrafa, Amaro, Chaves, Lourosa, Martins, & Mendes, 2005)
RussNet	Russian	(Azarova, Mitrofanova, Sinopalnikova, Yavorskaya, & Oparin, 2002)
Sanskrit WordNet	Sanskrit	(Kulkarni, Dangarikar, Kulkarni, Nanda, & Bhattacharyya, 2010)
Sinhala WordNet	Sinhala	(Welgama, Herath, Liyanage, Udalamatta, Weerasinghe, & Jayawardana, 2011)
Tamil WordNet	Tamil	(Thiyagarajan, Arulmozi, & Rajendran, 2002)

Table 18 - WordNets in the World

Dimension	Component	Description	Relevant tools
Ontology Management	Ontology repository	Stores and accesses ontologies and ontology instances	3Store, AllegroGraph, Corese, Hawk, Jena, Kowari, OWLIM, Sesame, Virtuoso Universal Server, 4store.
	Alignment repository	Stores and accesses alignments	Alignment Server, COMA++.
	Ontology metadata registry	Stores and accesses ontology metadata information	Oyster, SchemaWeb, Ontology Metadata Vocabulary.
	Ontology management APIs	Provide programming interfaces for managing ontologies and ontology instances	OWL API, RDF2Go, SemWeb.NET, Pubby, Elda.
Querying and Reasoning	Ontology reasoner	Takes care of reasoning over ontologies and ontology instances	CEL, Cerebra Engine, FaCT ++, fuzzyDL, HermiT, KAON2, MSPASS, Pellet, QuOnto, RacerPro, SHER, SoftFacts, TrOWL
	Semantic search	Takes care of the user interface for editing queries.	ARQ, Ginseng, K-Search, NLP-Reduce, Ontogator, PowerAqua, SemSearch.
	Ontology discover and ranking	Finds appropiate views, versions or subsets of ontologies and rank them according to some criterion	Swoogle, Watson, Sindice.
Ontology Engineering	Ontology editor	Allows creating and modifying ontologies, ontology elements and ontology documentation	DODDLE, graphl, GrOWL, ICOM, IsaViz, NeOn ToolKit, Ontotrack, Powl, Protégé, SemanticWorks, SemTalk, SWOOP, TopBraid Composer.
	Ontology browser	Allows to visually browse and ontology	Brownsauce, BrowseRDF, Disco, Fenfire, Jambalaya, Longwell, mSpace, OINK, Ontosphere 3D, Ontoviz, OWLViz, RDF Gravity, Tabulator, TGVizTab, Welkin.
	Ontology learner	Acquires knowledge and generates ontologies of a given domain through some kind of process.	KEA, OntoGen, OntoLearn, Text2Onto, TERMINAE.
	Ontology versioner	Mantains, stores and manages different versions of an ontology.	SemVersion
Ontology Processing	Ontology matcher	Matches two ontologies and outputs some alignments	AgreeMaker, AMW, AROMA, ASMOV, AUTOMS, CMS, CODI, COMA, Ef2Match, Falcon-AO, Gerome, HMatch, Lily, MapOnto, Mapso, OLA, OntoBuilder, PROMPT, RiMOM, S-Match, SAMBO.
	Ontology localization and profiling	Adapts an ontology according to some language, context or user profile.	LabelTranslator, lemmon editor.

	Ontology evaluator	Evaluates ontologies, either their formal model or their content in the different phases of their life cycle	CleOn, ConsVISor, Eyeball, VRP.
Instance Generation	Instance matcher	Is in charge of manual and semi-automatic matching of instances from different ontologies	SILK, LIMES
	Instance editor	Allows manually creating and modifying instances of concepts and/or relations	GATE, OCAT
	Manual annotation	Manual or semi-automatic annotation of digital content documents with concepts in the ontology	GATE, OCAT, OntoMat, Magpie, M-OntoMat, PhotoStuff
	Automatic annotation	Automatically annotates digital content with concepts in the ontology	KIM, GATE-ML
	Ontology populator	Automatically generates new instaces in a given ontology from a data source.	CLIE, NOR2O, R2O & ODEMapster, geometry2rdf.

Table 19 - Semantic technology dimensions